CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

APLICAÇÃO DE MINERAÇÃO DE DADOS NA DESCOBERTA DOS FATORES SOCIOECONÔMICOS ASSOCIADOS COM O DESEMPENHO DOS PARTICIPANTES DO ENEM

EDUARDO SOUZA COSTA ARAÚJO
HENRIQUE OLÍMPIO DE MEDEIROS SILVA

ANÁPOLIS
2020

EDUARDO SOUZA COSTA ARAÚJO
HENRIQUE OLÍMPIO DE MEDEIROS SILVA

APLICAÇÃO DE MINERAÇÃO DE DADOS NA DESCOBERTA DOS FATORES SOCIOECONÔMICOS ASSOCIADOS COM O DESEMPENHO DOS PARTICIPANTES DO ENEM

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a conclusão da disciplina de Trabalho de Conclusão de Curso II do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Profª. Esp. Aline Dayany de Lemos.

Anápolis
2020

Resumo

Realizado anualmente, o ENEM tem por objetivo avaliar o desempenho escolar ao fim do Ensino Médio. Além disso, a nota do exame permite aos participantes se candidatarem a vagas no Ensino Superior, em instituições públicas ou privadas. Porém as vagas são limitadas, e os resultados do participante são cruciais para seu sucesso neste objetivo. Ao realizar a inscrição e posteriormente o exame, os participantes fornecem dados referentes ao seu perfil socioeconômico, estes que podem ser aplicados na geração de conhecimento que revele a sua associação com o desempenho na prova. Diante deste cenário, buscou-se executar a atividade de Descoberta de Conhecimento em Bases de Dados, o que inclui a etapa de Mineração de Dados, nas bases de dados do ENEM dos anos de 2016 a 2018, a fim de obter a associação entre os fatores socioeconômicos e as notas dos participantes do exame em Goiás, utilizando o algoritmo de regras de associação *FP-Growth* e sendo implementado com o auxílio de bibliotecas *Python*.

Palavras-chave: Descoberta de Conhecimento em Bases de Dados, ENEM, Fatores Socioeconômicos, Mineração de Dados, Associação.

Abstract

Held annually, the ENEM aims to assess school performance at the end of High School. In addition, the exam grade allows the participants to apply for places in Higher Education, in public or private institutions. However, places are limited, and the participant's results are crucial to his success in this objective. When registering and then taking the exam, participants provide data on their socioeconomic profile, which can be used to generate knowledge that reveals their association with performance in the test. In view of this situation, we performed the activity of Knowledge Discovery in Databases, which includes the Data Mining stage, in the ENEM databases from 2016 to 2018, in order to obtain the association between the socioeconomic factors and the exam participants' grades from Goiás, using the FP-Growth association rules algorithm and being implemented using Python libraries.

Keywords: Knowledge Discovery in Databases, ENEM, Socioeconomic Factors, Data Mining, Association.

Lista de Figuras

Figura 1 – Estrutura Básica de Um Processo de DCBD 15
Figura 2 – Passos do Modelo de Processo KDD 17
Figura 3 – Passos do Modelo de Processo CRISP-DM 18
Figura 4 – Multidisciplinaridade da Mineração de Dados 21
Figura 5 – Matriz de Confusão da Classificação Binária 24
Figura 6 – Código para Execução da Tarefa de MD 36
Figura 7 – Seleção dos Atributos a Serem Utilizados 53
Figura 8 – Remoção dos Registros Incompletos 53
Figura 9 – Unificação das Bases de Dados 54
Figura 10 – Criação da Média das Notas das Provas Objetivas 54
Figura 11 – Categorização dos Atributos Numéricos 55
Figura 12 – Renomeação das Categorias 1 56
Figura 13 – Renomeação das Categorias 2 57
Figura 14 – Renomeação das Categorias 3 58
Figura 15 – Transformação das Categorias em Colunas 1 59
Figura 16 – Transformação das Categorias em Colunas 2 60
Figura 17 – Transformação das Categorias em Colunas 3 61

Lista de Quadros

Quadro 1 – Algoritmos para Mineração de Dados 27
Quadro 2 – Ferramentas para Mineração de Dados 29
Quadro 3 – *Hardware* Utilizado no Desenvolvimento 30
Quadro 4 – *Software* Utilizado no Desenvolvimento 30
Quadro 5 – Atributos Selecionados 32
Quadro 6 – Transformação do Atributo de Idade 34
Quadro 7 – Transformação da Média das Notas das Provas Objetivas 34
Quadro 8 – Transformação da Nota da Redação 34
Quadro 9 – Redução das Categorias de Pessoas na Residência (Q005) 35
Quadro 10 – Redução das Categorias de Renda (Q006) 35
Quadro 11 – Regras Geradas para as Notas Muito Baixas 37
Quadro 12 – Regras Geradas para as Notas Baixas 37
Quadro 13 – Regras Geradas para as Notas Regulares 38
Quadro 14 – Regras Geradas para as Notas Altas 39
Quadro 15 – Regras Geradas para as Notas Muito Altas 39

Lista de Tabelas

Tabela 1 – Total de Registros das Bases de Dados 31
Tabela 2 – Total de Registros Após o Pré-Processamento 33

Lista de Abreviaturas e Siglas

BD	Banco de Dados
CRISP-DM	*Cross-Industry Standard Process for Data Mining*
DCBD	Descoberta de Conhecimento em Bases de Dados
ENEM	Exame Nacional do Ensino Médio
FIES	Programa de Financiamento Estudantil
FN	Falso Negativo
FP	Falso Positivo
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
KDD	*Knowledge Discovery in Databases*
KDP	*Knowledge Discovery Process*
MD	Mineração de Dados
PDC	Processo de Descoberta de Conhecimento
PROUNI	Programa Universidade para Todos
RNA	Rede Neural Artificial
SGBD	Sistema Gerenciador de Banco de Dados
SISU	Sistema de Seleção Unificada
SQL	*Structured Query Language*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

Sumário

1 INTRODUÇÃO 11

2 FUNDAMENTAÇÃO TEÓRICA 13

2.1 Dado, Informação e Conhecimento 13

2.2 Banco de Dados 14

2.3 Descoberta de Conhecimento em Bases de Dados 14

2.3.1 Modelos de Processos de DCBD 16

2.3.2 Mineração de Dados 20

2.3.2.1 Tarefas 21

2.3.2.2 Técnicas 24

2.3.2.3 Algoritmos 27

2.3.2.4 Ferramentas 28

3 DESENVOLVIMENTO 30

3.1 Escolha da Ferramenta de MD 30

3.2 Ambiente de Desenvolvimento 30

3.3 Escolha do Processo de DCBD 30

3.4 Aplicação do Processo KDD 31

3.4.1 Compreensão do Domínio de Aplicação 31

3.4.2 Seleção dos Dados Alvo 31

3.4.3 Limpeza e Pré-Processamento dos Dados 31

3.4.4 Redução e Transformação dos Dados 33

3.4.5 Escolha da Tarefa de MD 35

3.4.6 Escolha do Algoritmo de MD 35

3.4.7 Execução da MD 36

3.4.8 Interpretação dos Resultados 40

4 CONSIDERAÇÕES FINAIS 41

4.1 Trabalhos Futuros 41

REFERÊNCIAS BIBLIOGRÁFICAS.....42

ANEXO A - DICIONÁRIO DE DADOS DO ENEM.....45

APÊNDICE A – LIMPEZA E PRÉ-PROCESSAMENTO DOS DADOS.....53

APÊNDICE B – REDUÇÃO E TRANSFORMAÇÃO DOS DADOS.....54

1 INTRODUÇÃO

O Exame Nacional do Ensino Médio (ENEM) busca avaliar o desempenho escolar ao final da Educação Básica. O exame é realizado anualmente desde 1998 e possibilita o acesso à Educação Superior através de programas como o Sistema de Seleção Unificada (SISU), o Programa Universidade para Todos (PROUNI) e o Programa de Financiamento Estudantil (FIES). Porém, a performance do participante é crucial para isso, pois as vagas são limitadas, permitindo a entrada apenas daqueles melhores classificados (INEP, 2019a).

Os dados dos participantes do ENEM são recolhidos através da inscrição e do exame, e atendem à demanda de informações específicas, tais como as questões das provas, os gabaritos, as informações sobre os itens, as notas e sobre o perfil socioeconômico dos inscritos (INEP, 2019b). Através destes dados e resultados do ENEM é possível a elaboração de estudos e indicadores educacionais (INEP, 2019a).

Segundo Júnior (2018), estes dados são disponibilizados pelo INEP e podem conter um conjunto de informações embutidas, que podem auxiliar, por exemplo, na compreensão dos fatores associados ao desempenho dos participantes que realizam o exame. E dentre estes fatores temos os socioeconômicos que, segundo Silveira, Barbosa e Silva (2015), são responsáveis por 66% da variância da nota dos participantes.

Porém, a fim de extrair conhecimento destes dados que possa auxiliar na compreensão dos fatores socioeconômicos que se relacionam com a performance dos participantes, é necessário a aplicação de uma atividade adequada para esse objetivo. Para tal, é proposto a aplicação da Mineração de Dados, executada através de um processo de Descoberta de Conhecimento, na base de dados do ENEM realizados em 2016, 2017 e 2018, entre os participantes de Goiás. Para a execução desta atividade, busca-se aplicar a tarefa, técnica e algoritmo que melhor se adequem aos dados obtidos e aos objetivos almejados, bem como as ferramentas para auxiliarem neste processo.

De acordo com Castro e Ferrari (2016), os avanços da tecnologia, tanto computacional quanto de comunicação, produzem um problema de superabundância de dados, devido a capacidade de coletá-los e armazená-los ser superior a habilidade de analisar e extrair conhecimento deles. Neste contexto, é necessário a aplicação de técnicas e ferramentas que transformem tais dados, de maneira inteligente e automática, em informações úteis que possam ser usadas em tomadas de decisão estratégicas.

Segundo Silva, Peres e Boscarioli (2016), considerando que as bases de dados são volumosas, e que o conhecimento pode ser implícito se torna necessário um trabalho de busca detalhado, ou metaforicamente, uma “mineração” destes dados.

Em relação ao ENEM, o exame conta com uma elevada quantidade de participantes, tendo em 2018 um total de 5,5 milhões de inscritos, com 4,1 milhões destes comparecendo aos dois dias de prova. Este número de participantes gera uma grande base de dados, tanto pessoais quanto sobre sua performance no exame (INEP, 2019c).

Conforme Júnior (2018), esta base de dados do ENEM contém informações que podem ser aplicadas na tomada de decisões estratégicas para o direcionamento de políticas educacionais promissoras, que auxiliem educadores e gestores na busca de ações visando à melhoria na qualidade do ensino.

Porém, a utilização dos dados de exames como o ENEM para compreender o ensino é uma área em estágios iniciais no Brasil, apesar do reconhecimento de sua importância (KLEINKE, 2017).

Baseando-se nessas informações, a Mineração de Dados foi escolhida como atividade para analisar os dados dos participantes goianos do ENEM aplicados entre 2016 e 2018, a fim de descobrir os fatores socioeconômicos, como renda, estrutura familiar, tipo de ensino e outros presentes no questionário respondido pelos inscritos, que estão associados ao seu desempenho no exame.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são abordados os tópicos necessários para a compreensão deste trabalho, apresentando uma explicação sobre dados, informação e conhecimento, banco de dados, Descoberta de Conhecimento em Bases de Dados e a etapa de Mineração de Dados e suas características.

2.1 Dado, Informação e Conhecimento

Segundo Silva, Peres e Boscarioli (2016), dado, informação e conhecimento se confundem como sinônimos, porém possuem diferentes definições no contexto da Mineração de Dados (MD).

O dado consiste em um fato, um valor documentado ou um resultado de uma medição, e é a matéria-prima para que os processos de mineração ocorram. Os dados podem ser entendidos como o nível mais básico de abstração, de onde a informação e, posteriormente, o conhecimento serão extraídos. Tais dados são organizados em coleções denominadas bases de dados, que permitem uma recuperação eficiente dos mesmos (CASTRO; FERRARI, 2016; SILVA; PERES; BOSCARIOLI, 2016).

Castro e Ferrari (2016) e Silva, Peres e Boscarioli (2016) definem que os dados podem ser encontrados em três formas:

- Estruturados: Os dados deste tipo estão organizados de alguma forma em campos fixos, como, por exemplo, em uma tabela ou em uma planilha. Os dados estruturados dependem de um modelo de dados que possua a descrição do seu conteúdo, com suas propriedades e relações.
- Não estruturados: São dados que não possuem um modelo e que não estão organizados de uma maneira predefinida, como por exemplo, textos, imagens e vídeos. Os dados não estruturados são habitualmente mais difíceis de indexar, acessar e analisar.
- Semiestruturados: Quando algum nível de organização é atribuído aos dados não estruturados, eles passam a ser denominados semiestruturados. Por exemplo, arquivos XML ou textos com *tags* (marcações).

Conforme Silva, Peres e Boscarioli (2016), ao atribuir aos dados um sentido semântico ou um significado, surge a informação. Quando estes significados são compreendidos, ou seja, um agente os aprende e se torna capaz de tomar decisões a fim de agregar valor a partir deles, surge o conhecimento. Tanto o dado, como a informação e o

conhecimento são elementos fundamentais para a MD e a Descoberta de Conhecimento em Base de Dados (DCBD).

2.2 Banco de Dados

De acordo com Alves (2014), um banco de dados (BD) é um conjunto de dados com um significado implícito. Um BD representa uma parte do mundo real, e é construído e povoado com dados que possuem um determinado objetivo, com usuários e aplicações desenvolvidas para manipulá-los.

Um BD possui uma fonte de origem dos dados, um grau de interação com eventos no mundo real e usuários ativamente interessados em seu conteúdo. Para que um BD seja preciso o tempo todo, é necessário que ele reflita as mudanças da parte do mundo real que ele representa o mais breve possível (ELMASRI; NAVATHE, 2011).

Para descrever os tipos de informações que serão armazenadas, é criado um modelo de BD. Este modelo é uma descrição formal da estrutura de um BD e geralmente são divididos no modelo conceitual, que é abstrato e descreve a estrutura independente de um Sistema Gerenciador de Banco de Dados (SGBD), e o modelo lógico, que representa a estrutura de acordo com o SGBD utilizado. (HEUSER, 2009).

Os BDs são comumente classificados de acordo com o seu modelo de dados. Embora o mais comum seja o relacional, há diversos outros disponíveis, como o orientado à objetos, armazém de chave-valor, grafos e orientado à documentos (DIANA; GEROSA, 2010; ALVES, 2014).

A fim de definir, recuperar e alterar um BD, os usuários utilizam ferramentas e programas denominadas SGBDs. Estes SGBDs são capazes de executar tarefas avançadas de gerenciamento, utilizando habitualmente uma linguagem para manipulação dos BDs como, no caso dos BDs relacionais, a *Structured Query Language* (SQL), ou outras linguagens dependendo do modelo utilizado (HEUSER, 2009; ALVES, 2014).

2.3 Descoberta de Conhecimento em Bases de Dados

De acordo com Cios et al. (2007), o Processo de Descoberta de Conhecimento (PDC), em inglês, *Knowledge Discovery Process* (KDP), também chamado de Descoberta de Conhecimento em Bases de Dados (DCBD), em inglês, *Knowledge Discovery in Databases* (KDD), busca um novo conhecimento em algum domínio de aplicação. Fayyad,

Piatetsky-Shapiro e Smyth (1996) descrevem a DCBD como um processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis nos dados.

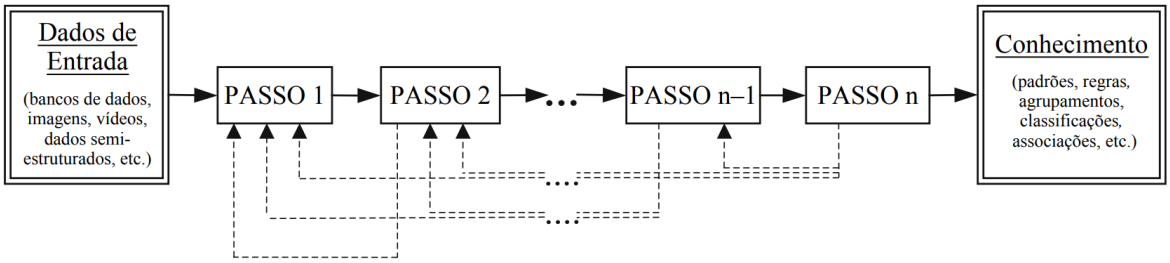
Conforme Silva, Peres e Boscarioli (2016), o processo de DCBD é analítico, sistemático e, até onde possível, automatizado, e tem como objetivo encontrar padrões intrínsecos nos dados, os apresentando de forma que a sua assimilação como conhecimento seja facilitada.

A primeira estrutura básica do modelo de DCBD foi proposta por Fayyad, Piatetsky-Shapiro e Smyth (1996) e posteriormente melhorada e modificada por outros (CIOS et al., 2007).

Nesta estrutura básica, conforme disposto na Figura 1, o processo de DCBD é formado por um conjunto de passos que os dados de entrada são submetidos para se obter conhecimento útil ao final, e pelos procedimentos que são realizados em cada um destes passos. Estes passos são executados em sequência, indo para o próximo quando o anterior for finalizado, com cada passo requerendo os resultados do anterior (CIOS et al., 2007).

Além disso, o processo de DCBD possui uma natureza iterativa, pois permite a repetição parcial ou integral de passos anteriores, a fim de obter resultados satisfatórios por meio de refinamentos sucessivos (GOLDSCHMIDT; PASSOS, 2005).

Figura 1 – Estrutura Básica de Um Processo de DCBD



Fonte: Adaptado de Cios et al. (2007, p. 11, tradução nossa)

Segundo Goldschmidt e Passos (2005), a aplicação do processo de DCBD envolve três tipos de componentes:

- O problema em que será aplicado, que é caracterizado pelo conjunto de dados que passarão pelo processo, o especialista no domínio da aplicação e os objetivos da aplicação;
- Os recursos disponíveis para a aplicação, que incluem o especialista em DCBD, as ferramentas e algoritmos (*software*), e a plataforma computacional (*hardware*);

- E os resultados obtidos, que compreendem o modelo de conhecimento gerado, que consiste em qualquer abstração de conhecimento, expresso em alguma linguagem, que descreva um conjunto de dados.

2.3.1 Modelos de Processos de DCBD

De acordo com Cios et al. (2007), os modelos de processo de DCBD começaram a surgir na década de 1990 através de pesquisas acadêmicas, seguidas rapidamente pela indústria.

Os esforços acadêmicos tinham como objetivo formular uma estrutura geral para o processo de MD que fosse aceita como um padrão, semelhante à como ocorreu com o SQL para os bancos de dados relacionais. Já o campo industrial buscou definir processos e metodologias que pudessem guiar a implementação da aplicação de MD (AZEVEDO; SANTOS, 2008).

Conforme Cios et al. (2007), as principais diferenças entre os modelos consistem no número e no escopo dos passos a serem seguidos. Como característica em comum, tem-se as entradas e saídas. As entradas geralmente são formadas por dados em vários formatos, como dados numéricos e nominais de bases de dados, imagens, vídeos, dados semiestruturados como XML e HTML, dentre outros. Já a saída é o conhecimento gerado, normalmente na forma de regras, padrões, associações etc.

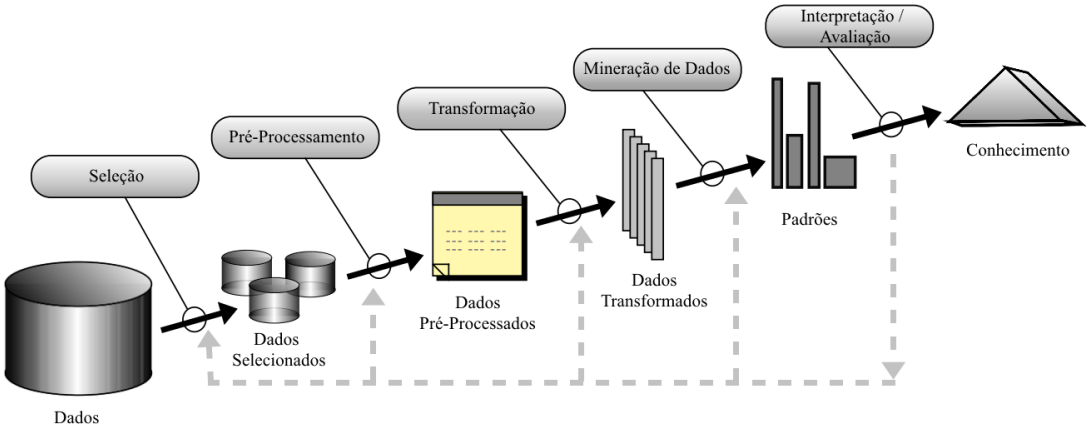
Os modelos acadêmicos geralmente não são desenvolvidos pensando em questões industriais, porém podem ser aplicados de forma relativamente fácil no âmbito industrial e vice-versa (CIOS et al., 2007).

a) *Knowledge Discovery in Databases* (KDD)

O KDD, modelo que dá o nome ao processo de DCBD, foi desenvolvido por Fayyad, Piatetsky-Shapiro e Smyth (1996), quando os esforços acadêmicos buscavam prover uma sequência de atividades para ajudar no processo de descoberta de conhecimento em qualquer domínio que fosse aplicado (CIOS et al., 2007).

O modelo possui nove passos, conforme a Figura 2, que segundo Fayyad, Piatetsky-Shapiro e Smyth (1996) são:

Figura 2 – Passos do Modelo de Processo KDD



Fonte: Adaptado de Fayyad, Piatetsky-Shapiro e Smyth (1996, p. 41, tradução nossa)

- Compreensão do domínio da aplicação: O primeiro passo inclui aprender sobre o negócio em que o KDD será aplicado e identificar os objetivos do ponto de vista do usuário.
- Seleção dos dados alvo: No segundo passo, o conjunto de dados a ser usado é escolhido, ou um subconjunto de variáveis ou amostras de dados.
- Limpeza e pré-processamento dos dados: No terceiro passo é realizado operações básicas como remover ruídos quando necessário, lidar com valores destoantes e definir estratégias de como tratar valores ausentes.
- Redução e transformação dos dados: O quarto passo consiste em encontrar atributos para representar os dados dependendo do objetivo escolhido. Através de redução dimensional ou métodos de transformação, o número de variáveis pode ser reduzido e os dados padronizados.
- Escolha da tarefa de Mineração de Dados: O quinto passo busca definir uma tarefa que corresponda os objetivos definidos no primeiro passo. Podendo ser classificação, regressão, agrupamento, dentre outros.
- Escolha do algoritmo de Mineração de Dados: No sexto passo, a fim de buscar por padrões nos dados, será selecionado o algoritmo (ou mais de um) mais adequado para tal atividade. Esse fase inclui decidir quais parâmetros são mais apropriados para o objetivo proposto.
- Mineração de Dados: O sétimo passo é onde os padrões nos dados serão buscados e representados de uma determinada forma, como regras de classificação, árvore de decisão, regressão, agrupamentos etc.

- Interpretação dos padrões: O oitavo passo envolve a visualização dos padrões ou modelos extraídos ou a visualização dos dados com estes modelos. Nesta etapa pode ocorrer um retorno aos passos anteriores para iteração adicional.
- Aplicação do conhecimento descoberto: O último passo consiste em agir a partir do conhecimento descoberto, usá-lo diretamente, incorporá-lo em outro sistema para ações adicionais ou documentar e reportar as partes interessadas. Esta fase também envolve verificar e resolver conflitos com conhecimentos previamente obtidos.

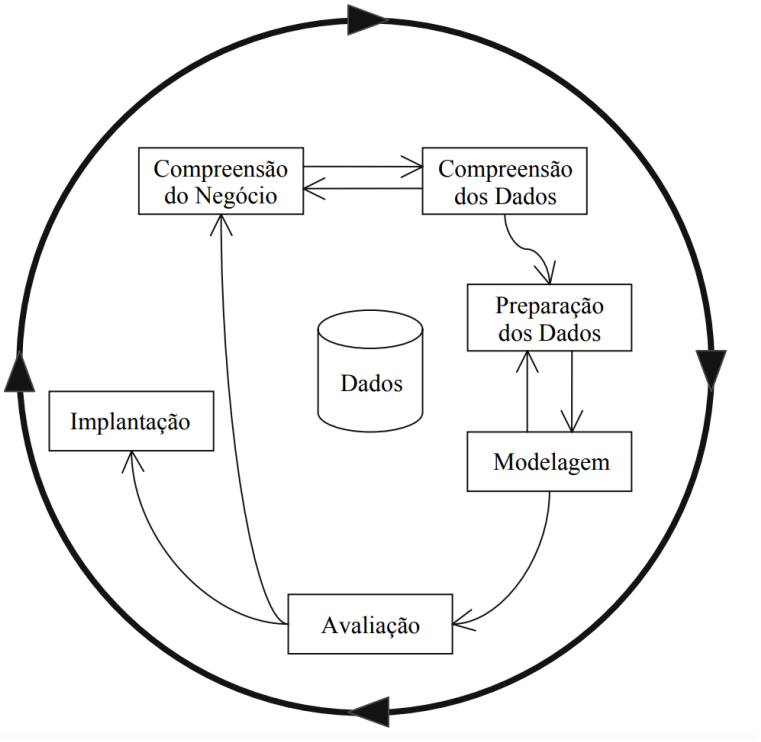
Segundo Cios et al. (2007), o KDD é o modelo mais popular e mais citado, fornece descrições técnicas detalhadas sobre análise de dados, porém lhe falta aspectos comerciais.

b) *Cross-Industry Standard Process for Data Mining* (CRISP-DM)

O CRISP-DM, ou Processo Padrão Inter-Industrial para Mineração de Dados, é um modelo desenvolvido no final da década de 1990 pelo consórcio de quatro empresas: *Integral Solutions Ltd*, NCR, *DaimlerChrysler* e OHRA, as duas últimas servindo como fonte de dados e estudo de caso (CIOS et al., 2007).

Conforme Chapman et al. (2000) o modelo é composto pelos seis passos, de acordo com a Figura 3, descritos a seguir:

Figura 3 – Passos do Modelo de Processo CRISP-DM



Fonte: Cios et al. (2007, p. 13, tradução nossa)

- Compreensão do negócio: O primeiro passo consiste em entender os objetivos e requisitos do projeto de uma perspectiva do negócio, transformar isto em um problema para a MD e criar um plano inicial para alcançar os objetivos.
- Compreensão dos dados: No segundo passo se inicia com a coleta dos dados e segue para atividades que possibilitam entendê-los, identificar problemas em sua qualidade, descobrir informações iniciais sobre estes dados e encontrar subconjuntos que permitam formar hipóteses sobre informações ocultas.
- Preparação dos dados: O terceiro passo cobre as atividades necessárias para formar o conjunto de dados a ser utilizado nos passos seguintes a partir dos dados obtidas na etapa anterior. Esta atividade inclui a seleção de tabelas, registros e atributos, assim como transformação e limpeza dos dados. Essa fase pode ser executada diversas vezes sem ordem específica.
- Modelagem: No quarto passo várias técnicas de modelagem são selecionadas e aplicadas, tendo seus parâmetros ajustados a fim de otimizar os valores. Normalmente há diversas técnicas para o mesmo problema de MD, algumas necessitando que os dados estejam em um formato específico. Retornar a fase de preparação pode ser necessário.
- Avaliação: No quinto passo, após construir um modelo (ou modelos) que apresente alta qualidade de um ponto de vista de análise de dados, é necessário avaliá-lo e verificar os passos executados, a fim de certificar de que o que foi feito atinge os objetivos do negócio.
- Implantação: Por último, o conhecimento obtido precisa ser organizado e apresentado de uma forma que o cliente ou usuário consiga usá-lo. Dependendo dos requisitos, a fase de implantação pode ser simples, gerando apenas um relatório, ou mais complexa, como implementar um processo contínuo de MD na empresa.

Conforme Cios et al. (2007) e Azevedo e Santos (2008) o CRISP-DM é bem documentado e todos os seus passos estão devidamente organizados e estruturados, permitindo que um projeto seja facilmente compreendido e revisado.

2.3.2 Mineração de Dados

Castro e Ferrari (2016) definem que o termo Mineração de Dados foi escolhido como um paralelo ao processo tradicional de mineração, onde se explora uma base de dados (mina), usando algoritmos (ferramentas), adequados para obter conhecimento (minerais preciosos).

A MD pode ser definida como um processo automático ou semiautomático de explorar grandes bases de dados, com o objetivo de descobrir padrões relevantes nos dados utilizados, e que sejam úteis para embasar a assimilação de informação importante, a fim de gerar conhecimento (SILVA; PERES; BOSCARIOLI, 2016).

O processo de MD é uma parte integrante da atividade de DCBD. Embora ambos sejam usados normalmente como sinônimos, foi proposto na primeira conferência internacional sobre DCBD, realizada em Montreal, Canadá, em 1995, que a terminologia Mineração de Dados fosse empregada apenas para a etapa de descoberta do processo de DCBD, que inclui outros passos como, por exemplo, seleção, pré-processamento, transformação e avaliação (CASTRO; FERRARI, 2016).

Fayyad, Piatetsky-Shapiro e Smyth (1996) afirmam que a MD é a aplicação de algoritmos específicos para extrair padrões dos dados, e que os outros processos da DCBD são tão essenciais quanto a etapa de mineração para garantir que conhecimento útil seja obtido, pois a aplicação apenas da MD pode ser perigosa, levando a descoberta de padrões inválidos e sem significado.

Castro e Ferrari (2016) descrevem a MD como interdisciplinar e multidisciplinar, pois envolve conhecimentos de diversas áreas. Goldschmidt e Passos (2005) destaca a relação com áreas como estatística, inteligência computacional e aprendizado de máquina, reconhecimento de padrões e banco de dados. A Figura 4 apresenta algumas das principais disciplinas envolvidas na MD:

Figura 4 – Multidisciplinaridade da Mineração de Dados

Fonte: Castro e Ferrari (2016, p. 7)

Camilo e Silva (2009) apresentam que a MD é aplicada de forma satisfatória em áreas como a eleitoral, telemarketing, bancária, medicinal, segurança, recursos humanos, turismo, dentre outras. Witten et al. (2016) relata que grande parte das aplicações de MD estão na área de comércio, domínio em que empresas possuem uma quantidade massiva de dados potencialmente valiosos.

A MD também é aplicável na área educacional, por exemplo, para identificar fatores que afetam a aprendizagem, desenvolver sistemas educacionais mais eficazes, ou verificar situações em que um tipo de abordagem instrucional proporciona melhores benefícios aos alunos (BAKER; ISOTANI; CARVALHO, 2011).

2.3.2.1 Tarefas

De acordo com Silva, Peres e Boscarioli (2016), dependendo do tipo de dado disponível e do tipo de conhecimento visado, há diferentes tipos de soluções e possibilidades. Para tal, a área de MD é dividida em tarefas, as quais auxiliam em como situar um problema real junto aos diferentes algoritmos disponíveis de análise de dados, e que tipo de padrão e conhecimento é possível descobrir.

As tarefas de MD são geralmente divididas em duas categorias, que conforme Castro e Ferrari (2016) são as descritivas, que caracterizam as propriedades dos gerais dos dados, e as preditivas, que fazem inferência a partir dos dados objetivando predições.

Nas tarefas descritivas, o objetivo é encontrar padrões que descrevem os dados de maneira que um agente humano possa interpretar. As análises descritivas permitem uma sumarização e compreensão dos objetos de uma base e de seus atributos (CASTRO; FERRARI, 2016; SILVA; PERES; BOSCARIOLI, 2016).

Segundo Camilo e Silva (2009) e Castro e Ferrari (2016), as tarefas preditivas visam descobrir o valor futuro de um determinado atributo. Na predição, uma parte dos dados disponíveis é usada na geração de um modelo preditivo (conjunto de treinamento), e outra parte é usada para avaliar a qualidade do modelo gerado (conjunto de teste).

As tarefas descritivas e preditivas são especializadas em outras tarefas, sendo as principais:

a) Associação:

A associação faz parte das tarefas descritivas. Silva, Peres e Boscarioli (2016) descrevem esta tarefa como a busca por ocorrências frequentes e simultâneas entre elementos de um contexto. Os algoritmos que resolvem esta tarefa analisam conjuntos de dados que representam eventos ou transações, procurando por itens que sejam frequentemente envolvidos nas mesmas situações ou que apresentam algum tipo de correlação em seus comportamentos. Espera-se deste tipo de tarefa que padrões inesperados sejam revelados, mesmo que seja comum a descoberta de padrões triviais.

Conforme Camilo e Silva (2009), a associação é uma das tarefas mais conhecidas devido aos bons resultados obtidos. Estes resultados gerados são apresentados na forma de: SE atributo X ENTÃO atributo Y.

b) Agrupamento:

O agrupamento (ou clusterização), também é integrante das tarefas descritivas. É utilizada para separar os registros de uma base de dados em subconjuntos (ou, do inglês, *clusters*), de forma que seus elementos compartilhem de propriedades comuns que os diferenciem dos elementos de outros subconjuntos (GOLDSCHMIDT; PASSOS, 2005).

De acordo com Silva, Peres e Boscarioli (2016), nesta tarefa não há a necessidade do uso da informação sobre qualquer tipo de rotulação dos dados. Os algoritmos aplicados nesta tarefa executam procedimentos que organizam os dados em grupos, de forma que a similaridade entre os dados de um grupo seja máxima, e entre dados de grupos diferentes seja mínima.

A diferença do agrupamento para a tarefa de classificação (apresentada adiante) consiste no fato de que no agrupamento não há a necessidade de que os registros sejam

previamente categorizados. Esta tarefa também não tem a pretensão de classificar, estimar ou predizer o valor de uma variável, apenas identificando os dados similares (CAMILO; SILVA, 2009).

c) Classificação:

A classificação consiste em uma tarefa preditiva e é um dos métodos mais populares e importantes da MD (GOLDSCHMIDT; PASSOS, 2005).

Silva, Peres e Boscarioli (2016) descrevem a tarefa de classificação como um processo no qual se determina um mapeamento capaz de indicar a qual classe pertence qualquer exemplar de um domínio sob análise, com base em um conjunto de dados previamente classificados.

A tarefa de classificação pode ser dividida em duas categorias, binária e multiclasse. Na categoria binária, quantidade de classes é igual a dois. Caso esse valor seja superior a dois, então é categorizado como classificação multiclasse (SILVA; PERES; BOSCARIOLI, 2016).

Nesta tarefa, quando cada registro possui um rótulo de classe ou um valor de saída que representa o resultado de registros anteriores, a análise buscará construir um modelo que possa ser usado para predizer a saída para novos registros cujas as classes ou valor de saída são desconhecidos (CASTRO; FERRARI, 2016).

Conforme Silva, Peres e Boscarioli (2016), durante um processo de avaliação de um classificador é útil analisar os erros cometidos pelo modelo construído. Dependendo da natureza dos erros, é possível entender se há classes onde o classificar enfrenta problemas para tratar ou descobrir por que não está respondendo adequadamente mesmo com uma alta acurácia. Para essa análise é aplicável uma matriz de confusão.

A matriz de confusão de um classificador busca oferecer um detalhamento do desempenho do modelo proposto, mostrando para cada classe o número de acertos em relação ao número de classificações indicadas pelo modelo (GOLDSCHMIDT; PASSOS, 2005).

Na matriz de confusão para um problema binário (conforme Figura 5), cada célula recebe uma identificação. As duas classes presentes nestes problemas são definidas como classe positiva e classe negativa (SILVA; PERES; BOSCARIOLI, 2016).

Figura 5 – Matriz de Confusão da Classificação Binária

		Classe Predita	
		positivo	***negativo***
Classe Esperada	***positivo***	*Verdadeiros positivos (VP)*	*Falsos negativos (FN)*
	negativo	*Falsos positivos (FP)*	*Verdadeiros negativos (VN)*

Fonte: Silva, Peres e Boscarioli (2016, p. 131)

Cada uma das células da matriz apresenta um significado, que pode indicar problemas maiores ou menores nos resultados do classificador. Segundo Silva, Peres e Boscarioli (2016), os significados destas células em uma matriz binária são:

- Verdadeiro positivo (VP): O exemplar foi corretamente classificado como pertencente à classe positiva.
- Falso positivo (FP): O exemplar foi erroneamente classificado como pertencente à classe positiva.
- Verdadeiro negativo (VN): O exemplar foi corretamente classificado como pertencente à classe negativa.
- Falso negativo (FN): O exemplar foi erroneamente classificado como pertencente à classe negativa.

d) Regressão:

A regressão consiste em outra tarefa preditiva. É similar a classificação, porém se aplica apenas a atributos numéricos. (GOLDSCHMIDT; PASSOS, 2005).

Em sua aplicação, a regressão é usada para estimar valores utilizando como base um conjunto de dados históricos. A solução para a tarefa de regressão pode ser obtida a partir de métodos estatísticos baseados em premissas e condições relacionadas com o tipo de distribuição dos dados, ou de técnicas de aprendizado indutivo, que não necessitam de informação prévia sobre o tipo de distribuição dos dados. (SILVA; PERES; BOSCARIOLI, 2016).

2.3.2.2 Técnicas

Goldschmidt e Passos (2005) definem as técnicas como qualquer teoria que possa fundamentar a implementação de uma tarefa de MD. Cada técnica possui suas peculiaridades e apresentam melhor resultado dependendo do tipo de dado utilizado, não existindo uma classificação única para sua escolha e aplicação (CAMILO; SILVA, 2009).

Dentre as principais técnicas de MD, tem-se:

a) Algoritmo Genético

Segundo Camilo e Silva (2009), os Algoritmos Genéticos são baseados na teoria da evolução. Nesta técnica, uma população inicial é geralmente definida de forma aleatória, e seguindo a lei do mais forte, uma nova população é gerada com base na atual, com os indivíduos passando por processos de troca genética e mutação. Este processo é repetido até atingir um critério de parada.

Os Algoritmos Genéticos são úteis na busca de soluções de problemas complexos que envolvem otimização. Estes problemas complexos são de difícil modelagem matemática ou com um número muito grande de possibilidades (GOLDSCHMIDT; PASSOS, 2005).

b) Árvore de Decisão

Segundo Silva, Peres e Boscarioli (2016), uma Árvore de Decisão consiste em uma estrutura formada por nós internos e nós folhas, organizados em um modelo hierárquico, da mesma forma que uma estrutura de dados do tipo árvore.

Conforme Castro e Ferrari (2016), cada nó interno corresponde a um teste de um atributo, cada ramo representa um resultado do teste e os nós folhas representam classes ou distribuições de classes.

Uma vez construída, a árvore pode ser usada para classificar um objeto de classe desconhecida. Para isso, é necessário testar os valores dos atributos na árvore e percorrê-la até se atingir um nó folha, que será uma classe predita para o objeto testado (CASTRO; FERRARI, 2016).

c) Classificador Bayesiano

O Classificador Bayesiano é uma técnica estatística baseada no teorema de Thomas Bayes. Segundo este teorema, é possível encontrar a probabilidade de um certo evento ocorrer, dada a probabilidade de um outro evento que já ocorreu. Esta técnica parte do princípio de que não exista relação de dependência entre os atributos. (CAMILO; SILVA, 2009).

Segundo Camilo e Silva (2009) e Castro e Ferrari (2016), um Classificador Bayesiano possui desempenho comparável a uma Rede Neural Artificial e a uma Árvore de Decisão, em alguns problemas, além de apresentar alta acurácia e velocidade de processamento quando aplicados em grandes bases de dados.

d) Rede Neural Artificial

Uma Rede Neural Artificial (RNA), consiste em uma técnica computacional que constrói um modelo matemático inspirado em um sistema neural biológico, que possui uma capacidade de aprendizado, generalização, associação e abstração. Ao ter os dados repetidamente apresentados, as RNAs tentam aprender padrões a partir deles, procurando por relacionamentos e construindo modelos automaticamente, que são corrigidos para diminuir seu próprio erro (GOLDSCHMIDT; PASSOS, 2005).

Camilo e Silva (2009) definem uma RNA como um conjunto de unidades de entrada e saída conectadas por camadas intermediárias, onde cada ligação possui um peso associado. Estes pesos são ajustados durante o processo de aprendizado para conseguir classificar corretamente um objeto.

As RNAs necessitam de um período de treinamento e ajustes finos dos parâmetros, além das dificuldades na interpretação da relação entre a entrada e a saída. Porém, as RNAs conseguem trabalhar de forma que não seja afetada com valores incorretos e possa identificar padrões para os quais nunca foram treinadas (CAMILO; SILVA, 2009).

e) Regras de Associação

A Descoberta de Regras de Associação consiste em uma técnica usada na construção de relações sob a forma de regras entre itens de uma base de dados transacional. Seu objetivo é encontrar regras fortes de acordo com alguma medida do grau de interesse da regra (CASTRO; FERRARI, 2016).

Está técnica possui dois passos, na primeira os dados são analisados para se obtenha os conjuntos de itens mais frequentes (ou *itemsets*). Na segunda, tais itens são usados para a geração das regras. As regras que são geradas apresentam a forma de SE condição ENTÃO conclusão (GOLDSCHMIDT; PASSOS, 2005; CAMILO; SILVA, 2009).

Na obtenção dos itens frequentes é utilizada uma medida chamada suporte, que diz quantas vezes um conjunto de itens aparece em relação ao total de transações. Ao definir um suporte mínimo, os conjuntos abaixo do valor escolhido são desconsiderados (SILVA; PERES; BOSCARIOLI, 2016).

Conforme Castro e Ferrari (2016), o suporte é uma medida importante, pois, de acordo com o valor definido, regras de conjuntos que aparecem com pouca frequência podem ser eliminadas.

Por sua vez, na fase de geração das regras é aplicada a medida de confiança, que diz quantas vezes a conclusão ocorre em relação a premissa (ou condição), e é dada pelo

suporte do conjunto de itens que formam a regra em relação a premissa dessa mesma regra. Da mesma forma que ocorre com o suporte, ao definir uma confiança mínima as regras abaixo do valor definido são descartadas (SILVA; PERES; BOSCARIOLI, 2016).

Os valores escolhidos para o suporte e confiança definirão as regras que farão parte do conjunto final de regras geradas. Deve-se ter em mente que, ao definir estes parâmetros, regras que possam ser do interesse dos envolvidos no processo de DCBD podem ser eliminadas, sendo necessário cuidado ao estipulá-los (CASTRO; FERRARI, 2016).

2.3.2.3 Algoritmos

Os algoritmos são a implementação das técnicas de MD. As principais abordagens destes algoritmos são as de aprendizado supervisionado e não supervisionado. A principal diferença entre estas abordagens está no fato de que os métodos não supervisionados não necessitam de um atributo alvo (GOLDSCHMIDT; PASSOS, 2005; CAMILO; SILVA, 2009).

Nos algoritmos de aprendizado supervisionado, os rótulos das classes dos dados de treinamento são conhecidos previamente e usados para ajustar o modelo de predição. Este modelo pode ser usado para predizer o rótulo dos exemplares de teste, aqueles que não fizeram parte do treinamento. (CASTRO; FERRARI, 2016; SILVA; PERES; BOSCARIOLI, 2016).

Conforme Goldschmidt e Passos (2005), os algoritmos de aprendizado não supervisionado não possuem a informação da saída desejada. Estes algoritmos partem dos dados, buscando formar relacionamentos entre eles.

Os algoritmos supervisionados são geralmente empregados para as tarefas de classificação e regressão, enquanto os não supervisionados são utilizados nas de associação e agrupamento (CAMILO; SILVA, 2009).

No Quadro 1, de acordo com Goldschmidt e Passos (2005), Camilo e Silva (2009), Castro e Ferrari (2016) e Silva, Peres e Boscarioli (2016), são apresentados os principais algoritmos referentes a cada técnica apresentada:

Quadro 1 – Algoritmos para Mineração de Dados

Técnicas	**Algoritmos**
Algoritmo Genético	Rule Evolver
Árvore de Decisão	C4.5, CART e ID3
Classificador Bayesiano	*Naïve Bayes*

Técnicas	Algoritmos
Regras de Associação	*Apriori*, DHP, *FP-Growth* e GSP
RNA	*Backpropagation*

Fonte: Goldschmidt e Passos (2005), Camilo e Silva (2009), Castro e Ferrari (2016) e Silva, Peres e Boscarioli (2016)

2.3.2.4 Ferramentas

Segundo Goldschmidt e Passos (2005) e Camilo e Silva (2009), diversas ferramentas, comerciais e *open source*, que implementam ambientes integrados foram desenvolvidas com o objetivo de facilitar a execução da MD, possibilitando o seu uso por profissionais de outras áreas.

Goldschmidt e Passos (2005), Camilo e Silva (2009), Castro e Ferrari (2016), Grus (2016), Silva, Peres e Boscarioli (2016) e McKinney (2018) apresentam algumas ferramentas populares para MD:

a) *Oracle Data Mining*

O *Oracle Data Mining* é um *software* de MD onde as atividades de DCBD ocorrem no mesmo ambiente do SGBD *Oracle*, provendo uma plataforma integrada simples, segura e escalável. Esta integração permite com que os dados possam passar pelo processo de DCBD sem a necessidade de serem extraídos previamente.

b) *Orange*

O *Orange* é uma ferramenta gratuita quer permite a construção visual, por meio de blocos e fluxogramas, de processos de análise e MD. O *software* também possui pacotes adicionais focados em diferentes áreas como bioinformática, mineração de textos e visualização de dados.

c) *Python*

O *Python* é uma linguagem de programação interpretada, tendo aplicações como criação de páginas *web* e análise de dados, possuindo bibliotecas para diversas aplicações, como a *Pandas* para manipulação de dados estruturados ou tabulares, a *NumPy* para operações com dados numéricos e a *Mlxtend* para aplicação de algoritmos de MD.

d) *RapidMiner*

O *RapidMiner* é um sistema de MD com versões gratuitas e pagas, que permite a construção visual, por meio de blocos e fluxogramas, de processos de análise e MD, assim como o *Orange*, podendo conectar-se a diferentes fontes de dados, como arquivos e

SGBD. Esta ferramenta além de conter diferentes algoritmos de MD, permite a criação de algoritmos pelo usuário.

e) *SAS Enterprise Miner*

O *SAS Enterprise Miner* é uma ferramenta paga de análise de dados que possui uma família de produtos, dentre eles o módulo para MD, que possui algoritmos de análise e recursos para o planejamento de ações e união de algoritmos.

f) *Weka 3*

O *Weka 3* é uma ferramenta de código aberto, que possui a implementação de diferentes tarefas de MD, além de funções para a execução de outros passos de DCBD. O *software* é desenvolvido em *Java*, podendo ser integrada com ambientes de desenvolvimento desta linguagem para uma maior personalização do processo de MD.

O Quadro 2 apresenta um comparativo entre essas ferramentas:

Quadro 2 – Ferramentas para Mineração de Dados

	Oracle Data Mining	***Orange***	***Python***	***Rapid Miner***	***SAS Enterprise Miner***	***Weka 3***
Open Source		X	X			X
Software proprietário	X			X	X	
Suporte à tarefa de Associação	X	X	X	X	X	X
Suporte à tarefa de Agrupamento	X	X	X	X	X	X
Suporte à tarefa Classificação	X	X	X	X	X	X
Suporte à tarefa de Regressão	X	X	X	X	X	X
Suporte à Aprendizado de Máquina (*Machine Learning*)		X	X	X	X	X
Treinamento Oficial Gratuito						X
Integração com ambientes de desenvolvimento em Java			X			X
API com suporte à algoritmos personalizados			X	X		
Bibliotecas com foco em análise de dados			X			

Fonte: Goldschmidt e Passos (2005), Camilo e Silva (2009), Castro e Ferrari (2016), Grus (2016), Silva, Peres e Boscarioli (2016), 1&1 Ionos (2017), McKinney (2018), Oracle (2019), Orange (2019)

3 DESENVOLVIMENTO

Esta seção contém o desenvolvimento deste trabalho e está dividida conforme o processo de DCBD escolhido. Embora apresentadas de forma linear, as etapas foram realizadas e revisitadas em diferentes momentos, devido a característica iterativa da atividade de DCBD.

3.1 Escolha da Ferramenta de MD

A fim de executar a atividade de DCBD, se faz necessário o uso de uma ferramenta para realizar o processo de MD. Para tal, a linguagem *Python*, em sua versão 3.7.2, foi selecionada. A escolha foi feita pois, dentre as ferramentas apresentadas (seção 2.3.2.4), foi a única gratuita capaz de lidar com os dados obtidos para este trabalho.

A linguagem *Python* possui bibliotecas como a *Pandas*, a *NumPy* e a *Mlxtend*, que serão úteis para todo o processo de DCBD, e não apenas na fase de MD.

3.2 Ambiente de Desenvolvimento

A execução deste trabalho foi realizada no ambiente computacional que possui a seguinte configuração (Quadro 3):

Quadro 3 – *Hardware* Utilizado no Desenvolvimento

Hardware	**Característica**
Processador	*Intel Pentium Dual-Core* E5700 3,00GHz
Memória RAM	8GB DDR3 800MHz

Fonte: Araújo e Silva (2020)

Além da ferramenta de MD escolhida, outros *softwares* foram empregados para o desenvolvimento deste projeto, conforme o Quadro 4:

Quadro 4 – *Software* Utilizado no Desenvolvimento

Hardware	**Função**
Xubuntu 18.04 LTS	Sistema operacional
Visual Studio Code	Software de edição de texto para desenvolvimento dos códigos em *Python*

Fonte: Araújo e Silva (2020)

3.3 Escolha do Processo de DCBD

Dos processos de DCBD apresentados (seção 2.3.1), o KDD e o CRISP-DM, verifica-se que ambos são viáveis de serem aplicados.

Por possuir descrições detalhadas sobre análise de dados, além de ter o objetivo de prover uma sequência de atividades para auxiliar a descoberta de conhecimento em

qualquer domínio que for aplicado, o processo KDD foi escolhido para a execução deste trabalho.

3.4 Aplicação do Processo KDD

3.4.1 Compreensão do Domínio de Aplicação

A fim de compreender o domínio que a mineração de dados será aplicada, a base de dados do ENEM neste caso, alguns pontos podem ser levantados:

- Os participantes são avaliados em provas de linguagens e códigos, ciências humanas, matemática, ciências da natureza e a redação, tendo cada uma delas uma nota de zero a 1000;
- Os participantes respondem um questionário com 27 itens sobre seu perfil socioeconômico.

Como estes detalhes em mente, decidiu-se verificar como os fatores socioeconômicos estão relacionados com o desempenho dos participantes do ENEM.

3.4.2 Seleção dos Dados Alvo

Os dados a serem utilizados, referentes as edições do ENEM realizadas em 2016, 2017 e 2018, foram obtidos no Portal de Dados Abertos do INEP, disponibilizados através dos arquivos microdados_enem_2016.csv, microdados_enem_2017.csv e microdados_enem_2018.csv. Neles, os atributos estão representados por colunas e os registros por linhas.

A Tabela 1 apresenta o total de registros existentes nestes arquivos.

Tabela 1 – Total de Registros das Bases de Dados

Ano	Arquivo	Quantidade
2016	microdados_enem_2016.csv	8.627.368
2017	microdados_enem_2017.csv	6.731.342
2018	microdados_enem_2018.csv	5.513.748
Total		20.872.458

Fonte: Araújo e Silva (2020)

Em relação aos atributos, descritos no Anexo A, o arquivo correspondente ao ano de 2016 possui um total de 166, enquanto o dos anos de 2017 e 2018 um total de 137.

3.4.3 Limpeza e Pré-Processamento dos Dados

Após a seleção dos dados, é realizado as atividades de limpeza e pré-processamento dos dados. Nesta etapa, foi utilizada a linguagem *Python* juntamente com sua biblioteca *Pandas.*

Inicialmente, foram selecionados os atributos a serem utilizados (Quadro 5), que correspondem a um total de 36. Foram desconsideradas as colunas duplicadas, com poucos registros, que estavam presentes em apenas uma das bases ou que não fossem aplicáveis ao objetivo proposto, como o número de inscrição, condições de aplicação da prova, dependência administrativa da escola e sua situação de funcionamento e perguntas do questionário descontinuadas a partir de 2017. O código utilizado para esta seleção é apresentado na Figura 7 (Apêndice A).

Quadro 5 – Atributos Selecionados

Atributo	**Tipo**
NU_ANO	Numérico
CO_MUNICIPIO_RESIDENCIA	Numérico
CO_UF_RESIDENCIA	Numérico
NU_IDADE	Numérico
TP_SEXO	Categórico
TP_COR_RACA	Categórico
NU_NOTA_CN	Numérico
NU_NOTA_CH	Numérico
NU_NOTA_LC	Numérico
NU_NOTA_MT	Numérico
NU_NOTA_REDACAO	Numérico
Q001	Categórico
Q002	Categórico
Q003	Categórico
Q004	Categórico
Q005	Categórico
Q006	Categórico
Q007	Categórico
Q008	Categórico
Q009	Categórico
Q010	Categórico
Q011	Categórico
Q012	Categórico
Q014	Categórico
Q015	Categórico
Q016	Categórico
Q017	Categórico
Q019	Categórico
Q020	Categórico

Atributo	Tipo
Q021	Categórico
Q022	Categórico
Q023	Categórico
Q024	Categórico
Q025	Categórico
Q026	Categórico
Q027	Categórico

Fonte: Araújo e Silva (2020)

Com exceção dos atributos “Q026” e “Q027” que aparecem, respectivamente, como “Q046” e “Q047” na base de dados de 2016, todos os demais são nomeados da mesma forma nos três arquivos.

Além disso, devido a limitações de memória do *hardware* empregado, foi necessário reduzir a quantidade de registros, para que o mesmo pudesse executar a tarefa de MD, para isso o escopo foi limitado aos participantes do estado de Goiás. Após este processo, todos os registros restantes possuíam o mesmo valor na coluna “CO_UF_RESIDENCIA”, então ela foi retirada.

Também foram removidos os registros dos inscritos que não realizaram o exame em sua totalidade (as quatro provas e a redação). Após isso, verificou-se que apenas três registros possuíam algum outro campo vazio, optou-se então por removê-los também. A Figura 8 (Apêndice A) mostra o código empregado nesta etapa, tendo seus resultados apresentados na Tabela 2.

Tabela 2 – Total de Registros Após o Pré-Processamento

Ano	Arquivo	Em Goiás	Com Todas as Notas	Sem Campos Vazios
2016	microdados_enem_2016.csv	286.980	191.815	191.812
2017	microdados_enem_2017.csv	219.996	142.928	142.928
2018	microdados_enem_2018.csv	191.029	133.609	133.609
Total		698.005	468.352	468.349

Fonte: Araújo e Silva (2020)

3.4.4 Redução e Transformação dos Dados

Para tornar o conjunto de dados adequado para a aplicação da tarefa de MD, certas alterações são necessárias. Primeiramente, as três bases de dados foram agrupadas em um único arquivo, através do código mostrado na Figura 9 (Apêndice B).

Após isso, para obter o desempenho geral dos participantes no exame, os atributos “NU_NOTA_CN”, “NU_NOTA_CH”, “NU_NOTA_LC”, “NU_NOTA_MT”, correspondentes as notas de cada competência da prova objetiva dos participantes do exame, foram substituídos por um novo, chamada “MED_SEM_RED”, onde foi armazenado a média desses valores (Figura 10 – Apêndice B).

Seguidamente, para padronizar o tipo de variável, as variáveis numéricas foram transformadas em variáveis categóricas (Figura 11 – Apêndice B). A idade foi classificada na forma apresentada no Quadro 6.

Quadro 6 – Transformação do Atributo de Idade

Faixa	**Categoria**
Abaixo de 21 anos	idade_menor_21
Entre 21 e 30 anos	idade_21_30
Entre 31 e 40 anos	idade_31_40
Acima de 41 anos	idade_maior_41

Fonte: Araújo e Silva (2020)

Os municípios, identificados por seu código, foram divididos em três categorias, sendo elas capital, região metropolitana e interior.

Também foram categorizadas as notas, tanto a média das provas objetivas (Quadro 7) quanto a redação (Quadro 8), com o mesmo critério aplicado pelo INEP (INEP, 2015).

Quadro 7 – Transformação da Média das Notas das Provas Objetivas

Faixa	**Categoria**
Abaixo de 450	nota_ob_mt_baixa
Entre 450 e 549,99	nota_ob_baixa
Entre 550 e 649,99	nota_ob_regular
Entre 650 e 749,99	nota_ob_alta
750 ou mais	nota_ob_mt_alta

Fonte: Araújo e Silva (2020)

Quadro 8 – Transformação da Nota da Redação

Faixa	**Categoria**
Abaixo de 500	nota_red_mt_baixa
Entre 500 e 599,99	nota_red_baixa
Entre 600 e 699,99	nota_red_regular
Entre 700 e 799,99	nota_red_alta
800 ou mais	nota_red_mt_alta

Fonte: Araújo e Silva (2020)

As demais colunas, que não necessitavam desta conversão, tiveram suas categorias reduzidas nos casos em que apresentavam mais de cinco categorias, para facilitar posteriormente na aplicação da tarefa de MD, como os atributos referentes a renda (Quadro 7) e a pessoas na residência (Quadro 8), além de serem renomeadas para que seus nomes refletissem o atributo e a categoria a qual pertencem (Figura 12 – Apêndice B).

Quadro 9 – Redução das Categorias de Pessoas na Residência (Q005)

Categoria Anterior	Nova Categoria
1	mora_sozinho
2 até 4	mora_2_a_4
5 até 7	mora_5_a_7
8 até 11	mora_8_a_11
11 ou mais	mora_mais_11

Fonte: Araújo e Silva (2020)

Quadro 10 – Redução das Categorias de Renda (Q006)

Categoria Anterior	Nova Categoria
A até D (até dois salários mínimos)	renda_ate_2_sal
E até G (entre dois e quatro salários mínimos)	renda_2_ate_4_sal
H até M (entre quatro e 10 salários mínimos)	renda_4_ate_10_sal
N até P (entre 10 e 20 salários mínimos)	renda_10_ate_20_sal
Q (20 ou mais salários mínimos)	renda_mais_20_sal

Fonte: Araújo e Silva (2020)

Por fim, todas as categorias foram transformadas em colunas, sendo representadas por zero nos registros em que não apareciam, e por um nos que estavam presentes. Nesta etapa, apresentada na Figura 15 (Apêndice B), além da biblioteca *Pandas* foi empregada a biblioteca *NumPy*.

3.4.5 Escolha da Tarefa de MD

Baseado nos objetivos propostos para este trabalho, foi escolhido como tarefa de mineração de dados a associação. Esta tarefa foi escolhida pois, dentre as tarefas de MD, é a capaz de gerar regras de associação entre os fatores socioeconômicos e as notas dos participantes ao aplicá-la nos dados obtidos.

3.4.6 Escolha do Algoritmo de MD

Dentre os algoritmos de descoberta de regras de associação existentes, foi escolhido o *FP-Growth*, pois, segundo Castro e Ferrari (2016) e Silva, Peres e Boscarioli (2016),

apresenta uma melhor resposta em relação a grandes conjuntos de dados, sem executar repetidas passagens pela base de dados, o que pode ser computacionalmente mais barato, diferente de algoritmos como o *Apriori*, que apresenta problemas em relação a estes fatores.

3.4.7 Execução da MD

Para executar a MD foi utilizada outra biblioteca *Python*, a *Mlxtend*, que fornece algoritmos de MD como o *FP-Growth*. O código empregado e o parâmetros de suporte e confiança utilizados são apresentados na Figura 6.

Figura 6 – Código para Execução da Tarefa de MD

```
import pandas as pd
from mlxtend.frequent_patterns import fpgrowth
from mlxtend.frequent_patterns import association_rules

dataset = pd.read_csv('pos_etapa6_enem.csv', sep = ';', encoding = 'utf-8')

frequent_itemsets = fpgrowth(dataset, min_support=0.35, use_colnames=True)
print (frequent_itemsets)

rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.35)
print (rules)

rules.to_csv('regras.csv', sep = ';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

As regras de associação foram geradas com foco em cada uma das categorias de nota definidas anteriormente (Quadro 7).

a) Nota muito baixa (abaixo de 450)

Conforme apresentado no Quadro 11, nas regras obtidas a partir dos casos em que a nota foi muito baixa, a com maior confiança, com 91%, foi do participante ter realizado o ensino médio em escola pública.

Também obteve-se regras com confiança superior a 50% que indicam os participantes nessa faixa de nota como tendo renda inferior a dois salários mínimos, não possuírem moto ou carro, receberem também nota muito baixa na redação, serem menores de 21 anos, possuírem acesso à internet mas não a um computador e morarem em um município do interior, com 2 a 4 pessoas.

Nas regras com confiança entre 35% e 50%, observa-se que a escolaridade de ambos os pais é de ensino fundamental incompleto, a residência possui dois quartos e um banheiro e a família possui três ou mais celulares.

Quadro 11 – Regras Geradas para as Notas Muito Baixas

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_ob_mt_baixa	ens_publica	91%
nota_ob_mt_baixa	renda_ate_2_sal	79%
nota_ob_mt_baixa	nota_red_mt_baixa	71%
nota_ob_mt_baixa	moto_nao	71%
nota_ob_mt_baixa	idade_menor_21	67%
nota_ob_mt_baixa	mora_2_a_4	66%
nota_ob_mt_baixa	mun_interior	63%
nota_ob_mt_baixa	internet_sim	59%
nota_ob_mt_baixa	carro_nao	58%
nota_ob_mt_baixa	comput_nao	51%
nota_ob_mt_baixa	quartos_dois, banheiro_um	45%
nota_ob_mt_baixa	esc_pai_ens_fund_inc	44%
nota_ob_mt_baixa	celular_tres_mais	39%
nota_ob_mt_baixa	esc_mae_ens_fund_inc	37%

Fonte: Araújo e Silva (2020)

b) Nota baixa (entre 450 e 550)

Em relação as notas baixas, observa-se no Quadro 12 que, com uma confiança de 82%, os participantes estudaram em escola pública.

Dentre as regras geradas com confiança superior a 50%, nota-se que os participantes possuem acesso à internet e a um computador, moram com duas a quatro pessoas, são menores de 21 anos, não possuem moto, a renda familiar é inferior a dois salários mínimos e residem em um município do interior.

Verifica-se nas regras com confiança entre 35% e 50% que o participante possui ensino médio completo, a família dispõe de três ou mais celulares, não possui carro, a residência conta com dois quartos e um banheiro e o pai possui o ensino fundamental incompleto.

Quadro 12 – Regras Geradas para as Notas Baixas

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_baixa	ens_publica	82%
nota_baixa	internet_sim	71%
nota_baixa	moto_nao	70%
nota_baixa	mora_2_a_4	69%
nota_baixa	idade_menor_21	68%
nota_baixa	renda_ate_2_sal	67%
nota_baixa	mun_interior	58%
nota_baixa	comput_um	53%
nota_baixa	carro_nao	49%
nota_baixa	celular_tres_mais	47%
nota_baixa	ens_med_concluido	46%
nota_baixa	quartos_dois, banheiro_um	41%

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_baixa	esc_pai_ens_fund_inc	40%

Fonte: Araújo e Silva (2020)

c) Nota regular (entre 550 e 650)

A partir das regras apresentadas no Quadro 13, verifica-se o acesso à internet dos participantes com nota regular com 86% de confiança.

Observa-se também, com confiança superior a 50%, que os participantes possuem menos de 21 anos, estudaram em escola pública, moram com duas a quatro pessoas, não possuem moto, detêm três ou mais celulares, um computador, a residência possui três ou mais quartos e residem na capital.

Nas regras entre 35% e 50%, nota-se que já concluíram o ensino médio, a renda familiar é inferior a dois salários mínimos e a escolaridade da mãe consiste no ensino médio completo.

Quadro 13 – Regras Geradas para as Notas Regulares

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_regular	internet_sim	86%
nota_regular	mora_2_a_4	73%
nota_regular	moto_nao	73%
nota_regular	idade_menor_21	72%
nota_regular	comput_um	60%
nota_regular	celular_tres_mais	61%
nota_regular	mun_capital	59%
nota_regular	ens_publica	56%
nota_regular	quartos_tres_mais	55%
nota_regular	carro_um	51%
nota_regular	ens_med_concluido	49%
nota_regular	renda_ ate_2_sal	39%
nota_regular	esc_mae_ens_med_comp	36%

Fonte: Araújo e Silva (2020)

d) Nota alta (entre 650 e 750)

Conforme mostrado no Quadro 14, o acesso à internet entre aqueles com nota alta possui uma confiança de 96%.

Nas regras com confiança superior a 50%, nota-se que os participantes são menores de 21 anos, estudaram em uma instituição privada sem bolsa, concluíram o ensino médio, moram com duas a quatro pessoas, não possuem moto, detêm três ou mais celulares, um computador, residem na capital e a mãe possui o ensino superior completo.

Com 35% a 50% de confiança, verifica-se que o pai possui o ensino superior completo, a família dispõe de um carro e a renda está entre quatro e 10 salários mínimos.

Quadro 14 – Regras Geradas para as Notas Altas

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_alta	internet_sim	96%
nota_alta	moto_nao	82%
nota_alta	idade_menor_21	77%
nota_alta	mora_2_a_4	76%
nota_alta	celular_tres_mais	74%
nota_alta	quartos_tres_mais	72%
nota_alta	ens_priv_sem_bolsa	65%
nota_alta	mun_capital	59%
nota_alta	ens_med_concluido	58%
nota_alta	esc_mae_ens_sup_comp	57%
nota_alta	comput_um	51%
nota_alta	carro_um	48%
nota_alta	esc_pai_ens_sup_comp	44%
nota_alta	renda_4_ate_10_sal	39%

Fonte: Araújo e Silva (2020)

e) Nota muito alta (acima de 750)

Verifica-se no Quadro 15 que, com 97% de confiança, aqueles que obtiveram nota muito alta possuem acesso à internet.

Com confiança superior a 50%, nota-se que os participantes possuem menos de 21 anos, já concluíram o ensino médio, estudaram em uma instituição privada, não possuem moto, detêm três ou mais celulares, moram com duas a quatro pessoas, a casa possui três ou mais quartos, residem na capital e ambos os pais concluíram o ensino superior.

Nas regras com 35% a 50% de confiança, observa-se que a renda está entre quatro e 10 salários mínimos, possuem um computador e dois carros.

Quadro 15 – Regras Geradas para as Notas Muito Altas

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_mt_alta	internet_sim	97%
nota_mt_alta	moto_nao	86%
nota_mt_alta	idade_menor_21	85%
nota_mt_alta	ens_priv_sem_bolsa	85%
nota_mt_alta	celular_tres_mais	80%
nota_mt_alta	mora_2_a_4	78%
nota_mt_alta	quartos_tres_mais	78%
nota_mt_alta	mun_capital	75%
nota_mt_alta	esc_mae_ens_sup_comp	75%
nota_mt_alta	esc_pai_ens_sup_comp	67%
nota_mt_alta	ens_med_concluido	61%

Antecedente (SE)	Consequência (ENTÃO)	Confiança
nota_mt_alta	carro_dois	44%
nota_mt_alta	renda_4_ate_10_sal	37%
nota_mt_alta	comput_um	37%

Fonte: Araújo e Silva (2020)

3.4.8 Interpretação dos Resultados

Após etapa de mineração das regras de associação, alguns resultados podem ser observados:

- A confiança da regra em relação ao acesso à internet aumenta junto com as notas, sendo de 59% para as notas muito baixas e de 97% para as muito altas;
- Aqueles com nota muito baixa possuem 39% de confiança de terem três celulares ou mais, contra 80% dos com nota mais altas;
- Nos computadores a confiança de quem tirou nota muito baixa não possuir um computador é de 51%, enquanto aqueles com nota muito alta tem 37% de ter apenas um;
- No tipo de ensino, a educação pública aparece nas notas mais baixas com 91% de confiança e decaindo nas demais notas, o oposto ocorre com a educação privada, que chega a 85% de confiança nas notas mais altas;
- Na renda familiar, os participantes com notas muito baixas possuem uma confiança de 71% de terem uma renda inferior a dois salários mínimos, contra 37% da renda entre quatro e 10 salários mínimos daqueles com nota muito alta;
- A escolaridade dos pais dos participantes é maior entre aqueles com notas mais altas. Nas notas muito baixas, a confiança do pai e da mãe possuírem ensino fundamental incompleto é de, respectivamente, 44% e 37%, contra 75% e 67% da mãe e do pai daqueles com notas muito altas terem completado o ensino superior;
- A confiança de aqueles com nota muito baixa não possuir carro é 58%, enquanto entre os participantes que obtiveram nota muito alta é 44% de possuírem dois carros.

4 CONSIDERAÇÕES FINAIS

A DCBD é um processo com múltiplas etapas, o que inclui a atividade de MD, e permite encontrar novos padrões em conjuntos de dados. É um processo que pode ser aplicado em diversas áreas, o que inclui a educacional, e os conhecimentos descobertos podem ser usados para guiar tomadas de decisão.

Buscou-se com esse trabalho obter a associação entre fatores socioeconômicos e o desempenho dos participantes do ENEM, aplicando regras de associação com o algoritmo *FP-Growth*, e utilizando para isso a linguagem *Python* juntamente com suas bibliotecas *Pandas*, *NumPy*, e *Mlxtend*.

Embora tenha sido executada a tarefa de associação, durante o desenvolvimento desta pesquisa observou-se que a base de dados do ENEM permite a aplicação de outras tarefas, como a classificação ou agrupamento, o que dependerá do conhecimento visado.

Por último, espera-se que este trabalho posso estimular o emprego de DCBD no âmbito educacional, a fim de guiar tomadas de decisão assertivas que possam aprimorar a qualidade do ensino.

4.1 Trabalhos Futuros

São propostos como trabalhos futuros:

- Emprego deste processo nos dados dos demais estados brasileiros e comparação entre estados e regiões;
- Execução de outras tarefas de MD na base de dados do ENEM, como a classificação para predizer a nota.

REFERÊNCIAS BIBLIOGRÁFICAS

1&1 IONOS. ***Data Mining Tools for Better Data Analysis***. 2017. Disponível em: <https://www.1and1.com/digitalguide/online-marketing/web-analytics/a-comparison-of-data-mining-tools/>. Acesso em: 01 de nov. 2019.

ALVES, William Pereira. **Banco de Dados**. São Paulo: Érica, 2014.

AZEVEDO, Ana; SANTOS, Manuel Filipe. ***KDD, SEMMA and CRISP-DM: A Parallel Overview***. Amsterdã: *IADIS European Conference on Data Mining 2008*. [s.n.], 2008. p. 182 - 185. Disponível em: <https://pdfs.semanticscholar.org/7dfe/3bc6035da527deaa72007a27cef94047a7f9.pdf>. Acesso em: 10 de out. 2019.

BAKER, Ryan; ISOTANI, Seiji; CARVALHO, Adriana. **Mineração de Dados Educacionais: Oportunidades para o Brasil**. Revista Brasileira de Informática na Educação, [s.l.], v. 19, n. 02, p. 3-13, 2011. Sociedade Brasileira de Computacao - SB. http://dx.doi.org/10.5753/rbie.2011.19.02.03. Acesso em: 12 de out. 2019.

CAMILO, Cássio Oliveira; SILVA, João Carlos da. Mineração de Dados: **Conceitos, Tarefas, Métodos e Ferramentas**. Goiânia: Instituto de Informática - Universidade Federal de Goiás, 2009. Disponível em: <http://www.portal.inf.ufg.br/sites/default/files/uploads/relatoriostecnicos/RT-INF_001-09.pdf>. Acesso em: 21 out. 2019.

CASTRO, Leandro Nunes de; FERRARI, Daniel Gomes. **Introdução à Mineração de Dados: Conceitos Básicos, Algoritmos e Aplicações**. São Paulo: Saraiva, 2016.

CHAPMAN, Pete et al. ***CRISP-DM 1.0: Step-by-Step Data Mining Guide***. 2000. Disponível em: <https://www.the-modeling-agency.com/crisp-dm.pdf>. Acesso em: 29 de out. 2019.

CIOS, Krzysztof J. et al. ***Data Mining: A Knowledge Discovery Approach***. New York: Springer, 2007.

DIANA, Mauricio de; GEROSA, Marco Aurélio. **NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0**. Departamento de Ciência da Computação - Universidade de São Paulo, 2010. Disponível em: <https://www.ime.usp.br/~mdediana/nosql_wtdbd10.pdf> Acesso em: 01 de nov. 2019.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. ***From Data Mining to Knowledge Discovery in Databases***. AI Magazine, v. 17, n. 3, p. 37-54, 1996. Disponível em: <https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1230/1131>. Acesso em: 05 de out. 2019.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. ***Data Mining*: Um Guia Prático - Conceitos, Técnicas, Algoritmos, Orientações e Aplicações**. Rio de Janeiro: Elsevier, 2005.

GRUS, Joel. **Data Science do Zero: Primeiras Regras com o Python**. Rio de Janeiro: Alta Books, 2016.

HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 6. ed. Porto Alegre: Bookman, 2009.

INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira). **ENEM**. 2019. Disponível em: <http://portal.inep.gov.br/web/guest/Enem> Acesso em: 26 de out. 2019.

INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira). **Nota Explicativa ENEM 2015 por Escola**. 2015. Disponível em: <http://download.inep.gov.br/educacao_basica/enem/nota_tecnica/2015/nota_explicativa_enem2015_por_escola.pdf> Acesso em: 02 de abril 2020.

INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira). **Resultados**. 2019. Disponível em: <http://portal.inep.gov.br/web/guest/educacao-basica/Enem/resultados> Acesso em: 21 de ago. 2019.

INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira). **Resultados do Enem 2018 já Foram Visualizados por 3 Milhões de Participantes**. 2019. Disponível em: <http://inep.gov.br/artigo/-/asset_publisher/B4AQV9zFY7Bv/content/resultados-do-enem-2018-ja-foram-visualizados-por-3-milhoes-de-participantes/21206> Acesso em: 02 de nov. 2019.

JÚNIOR, Wilton Moreira de Santana. **Mineração em dados do ENEM para a predição do desempenho acadêmico no âmbito da Rede Federal de Educação Tecnológica**. Recife: Universidade Federal de Pernambuco, 2018. Disponível em: <https://repositorio.ufpe.br/bitstream/123456789/29994/1/DISSERTA%c3%87%c3%83O%20Wilton%20Moreira%20de%20Santana%20J%c3%banior.pdf > Acesso em: 07 de dez. 2019.

KLEINKE, Maurício Urban. **Influência do status socioeconômico no desempenho dos estudantes nos itens de física do Enem 2012**. [S.l.]. Revista Brasileira de Ensino de

Física v. 39, n. 2, 2017. Disponível em: <http://www.scielo.br/pdf/rbef/v39n2/1806-1117-rbef-39-02-e2402.pdf> Acesso em: 07 de dez. 2019.

ORACLE. ***Oracle Data Mining: Scalable In-Database Predictive Analytics***. 2019. Disponível em: <https://www.oracle.com/database/technologies/advanced-analytics/odm.html>. Acesso em: 01 nov. 2019.

ORANGE. ***Orange Features***. 2019. Disponível em: <https://orange.biolab.si/#Orange-Features>. Acesso em 01. nov. 2019.

SILVA, Leandro Augusto da; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à Mineração de Dados: Com aplicações em R**. Rio de Janeiro: Elsevier, 2016.

SILVEIRA, Fernando Lang da; BARBOSA, Marcia Cristina Bernardes; SILVA, Roberto da. **Exame Nacional do Ensino Médio (ENEM): uma análise crítica**. [S.l.]. Revista Brasileira de Ensino de Física, v. 37, n. 1, 2015. Disponível em: <http://www.scielo.br/pdf/rbef/v37n1/1806-1117-rbef-S1806-11173710001.pdf> Acesso em: 07 de dez. 2019.

MCKINNEY, Wes. ***Python for Data Analysis: Data Wrangling with Pandas, NumPy and IPython***. 2. ed. Sebastopol: O'Reilly Media, 2018.

WITTEN, Ian H. et al. ***Data Mining: Pratical Machine Learning Tools and Techniques***. 4. ed. Cambridge: Morgan Kaufmann, 2016.

ANEXO A - DICIONÁRIO DE DADOS DO ENEM

Dicionário de Atributos - ENEM		
Nome	**Descrição**	**Tipo**
Dados do Participante		
NU_INSCRICAO	Número de inscrição	Numérico
NU_ANO	Ano do Enem	Numérico
CO_MUNICIPIO_RESIDENCIA	Código do município de residência 1º dígito: Região 1º e 2º dígitos: UF 3º, 4º, 5º e 6º dígitos: Município 7º dígito: dígito verificador	Numérico
NO_MUNICIPIO_RESIDENCIA	Nome do município de residência	Alfanumérico
CO_UF_RESIDENCIA	Código da Unidade da Federação de residência	Numérico
SG_UF_RESIDENCIA	Sigla da Unidade da Federação de residência	Alfanumérico
SG_UF_RESIDENCIA	Sigla da Unidade da Federação de residência	Alfanumérico
NU_IDADE	Idade	Numérico
TP_SEXO	Sexo	Categórico
TP_ESTADO_CIVIL	Estado Civil	Categórico
TP_COR_RACA	Cor/raça	Categórico
TP_NACIONALIDADE	Nacionalidade	Categórico
CO_MUNICIPIO_NASCIMENTO	Código do município de nascimento 1º dígito: Região 1º e 2º dígitos: UF 3º, 4º, 5º e 6º dígitos: Município 7º dígito: dígito verificador	Numérico
NO_MUNICIPIO_NASCIMENTO	Nome do município de nascimento	Alfanumérico
CO_UF_NASCIMENTO	Código da Unidade da Federação de nascimento	Numérico
SG_UF_NASCIMENTO	Sigla da Unidade da Federação de nascimento	Alfanumérico
TP_ST_CONCLUSAO	Situação de conclusão do Ensino Médio	Categórico
TP_ANO_CONCLUIU	Ano de Conclusão do Ensino Médio	Categórico
TP_ESCOLA	Tipo de escola do Ensino Médio	Categórico

TP_ENSINO	Tipo de instituição que concluiu ou concluirá o Ensino Médio	Categórico
IN_TREINEIRO	Indica se o inscrito fez a prova com intuito de apenas treinar seus conhecimentos	Categórico
Dados da Escola		
CO_ESCOLA	Código da Escola	Numérico
CO_MUNICIPIO_ESC	Código do município da escola 1º dígito: Região 1º e 2º dígitos: UF 3º, 4º, 5º e 6º dígitos: Município 7º dígito: dígito verificador	Numérico
NO_MUNICIPIO_ESC	Nome do município da escola	Alfanumérico
CO_UF_ESC	Código da Unidade da Federação da escola	Numérico
SG_UF_ESC	Sigla da Unidade da Federação da escola	Alfanumérico
TP_DEPENDENCIA_ADM_ESC	Dependência administrativa (Escola)	Categórico
TP_LOCALIZACAO_ESC	Localização (Escola)	Categórico
TP_SIT_FUNC_ESC	Situação de funcionamento (Escola)	Categórico
Dados dos Pedidos de Atendimento Especializado		
IN_BAIXA_VISAO	Indicador de baixa visão	Categórico
IN_CEGUEIRA	Indicador de cegueira	Categórico
IN_SURDEZ	Indicador de surdez	Categórico
IN_DEFICIENCIA_AUDITIVA	Indicador de deficiência auditiva	Categórico
IN_SURDO_CEGUEIRA	Indicador de surdo-cegueira	Categórico
IN_DEFICIENCIA_FISICA	Indicador de deficiência física	Categórico
IN_DEFICIENCIA_MENTAL	Indicador de deficiência mental	Categórico
IN_DEFICIT_ATENCAO	Indicador de déficit de atenção	Categórico
IN_DISLEXIA	Indicador de dislexia	Categórico
IN_DISCALCULIA	Indicador de discalculia	Categórico
IN_AUTISMO	Indicador de autismo	Categórico
IN_VISAO_MONOCULAR	Indicador de visão monocular	Categórico
IN_OUTRA_DEF	Indicador de outra deficiência ou condição	Categórico

	especial	
Dados dos Pedidos de Atendimento Específico		
IN_GESTANTE	Indicador de gestante	Categórico
IN_LACTANTE	Indicador de lactante	Categórico
IN_IDOSO	Indicador de inscrito idoso	Categórico
IN_ESTUDA_CLASSE_HOSPITALAR	Indicador de inscrição em Unidade Hospitalar	Categórico
Dados dos Pedidos de Recursos Especializados e Específicos para Realização das Provas		
IN_SEM_RECURSO	Indicador de inscrito que não requisitou nenhum recurso	Categórico
IN_BRAILLE	Indicador de solicitação de prova em braille	Categórico
IN_AMPLIADA_24	Indicador de solicitação de prova superampliada com fonte tamanho 24	Categórico
IN_AMPLIADA_18	Indicador de solicitação de prova ampliada com fonte tamanho 18	Categórico
IN_LEDOR	Indicador de solicitação de auxílio para leitura (ledor)	Categórico
IN_ACESSO	Indicador de solicitação de sala de fácil acesso	Categórico
IN_TRANSCRICAO	Indicador de solicitação de auxílio para transcrição	Categórico
IN_LIBRAS	Indicador de solicitação de Tradutor- Intérprete Libras	Categórico
IN_LEITURA_LABIAL	Indicador de solicitação de Tradutor- Intérprete Libras	Categórico
IN_MESA_CADEIRA_RODAS	Indicador de solicitação de mesa para cadeira de rodas	Categórico
IN_MESA_CADEIRA_SEPARADA	Indicador de solicitação de mesa e cadeira separada	Categórico
IN_APOIO_PERNA	Indicador de solicitação de apoio de perna e pé	Categórico
IN_GUIA_INTERPRETE	Indicador de solicitação de guia intérprete	Categórico
IN_COMPUTADOR	Indicador de solicitação de computador	Categórico
IN_CADEIRA_ESPECIAL	Indicador de solicitação de cadeira especial	Categórico
IN_CADEIRA_CANHOTO	Indicador de solicitação de cadeira para canhoto	Categórico
IN_CADEIRA_ACOLCHOADA	Indicador de solicitação de cadeira acolchoada	Categórico
IN_PROVA_DEITADO	Indicador de solicitação para	Categórico

	fazer prova deitado em maca ou mobiliário similar	
IN_MOBILIARIO_OBESO	Indicador de solicitação de mobiliário adequado para obeso	Categórico
IN_LAMINA_OVERLAY	Indicador de solicitação de protetor auricular	Categórico
IN_PROTETOR_AURICULAR	Indicador de solicitação de protetor auricular	Categórico
IN_MEDIDOR_GLICOSE	Indicador de solicitação de medidor de glicose e/ou aplicação de insulina	Categórico
IN_MAQUINA_BRAILE	Indicador de solicitação de máquina Braile e/ou Reglete e Punção	Categórico
IN_SOROBAN	Indicador de solicitação de soroban	Categórico
IN_MARCA_PASSO	Indicador de solicitação de marca-passo (impeditivo de uso de detector de metais)	Categórico
IN_SONDA	Indicador de solicitação de sonda com troca periódica	Categórico
IN_MEDICAMENTOS	Indicador de solicitação de medicamentos	Categórico
IN_SALA_INDIVIDUAL	Indicador de solicitação de sala especial individual	Categórico
IN_SALA_ESPECIAL	Indicador de solicitação de sala especial até 20 participantes	Categórico
IN_SALA_ACOMPANHANTE	Indicador de solicitação de sala reservada para acompanhantes	Categórico
IN_MOBILIARIO_ESPECIFICO	Indicador de solicitação de mobiliário específico	Categórico
IN_MATERIAL_ESPECIFICO	Indicador de solicitação de material específico	Categórico
IN_NOME_SOCIAL	Indicador de inscrito que se declarou travesti, transexual ou transgênero e solicitou atendimento pelo Nome Social, conforme é reconhecido socialmente em consonância com sua identidade de gênero	Categórico
Dados do Local de Aplicação da Prova		
CO_MUNICIPIO_PROVA	Código do município da aplicação da prova 1º dígito: Região	Numérico

	1º e 2º dígitos: UF 3º, 4º, 5º e 6º dígitos: Município 7º dígito: dígito verificador	
NO_MUNICIPIO_PROVA	Nome do município da aplicação da prova	Alfanumérico
CO_UF_PROVA	Código da Unidade da Federação da aplicação da prova	Alfanumérico
SG_UF_PROVA	Sigla da Unidade da Federação da aplicação da prova	Alfanumérico
Dados da Prova Objetiva		
TP_PRESENCA_CN	Presença na prova objetiva de Ciências da Natureza	Categórico
TP_PRESENCA_CH	Presença na prova objetiva de Ciências Humanas	Categórico
TP_PRESENCA_LC	Presença na prova objetiva de Linguagens e Códigos	Categórico
TP_PRESENCA_MT	Presença na prova objetiva de Matemática	Categórico
CO_PROVA_CN	Código do tipo de prova de Ciências da Natureza	Categórico
CO_PROVA_CH	Código do tipo de prova de Ciências Humanas	Categórico
CO_PROVA_LC	Código do tipo de prova de Linguagens e Códigos	Categórico
CO_PROVA_MT	Código do tipo de prova de Matemática	Categórico
NU_NOTA_CN	Nota da prova de Ciências da Natureza	Numérico
NU_NOTA_CH	Nota da prova de Ciências Humanas	Numérico
NU_NOTA_LC	Nota da prova de Linguagens e Códigos	Numérico
NU_NOTA_MT	Nota da prova de Matemática	Numérico
TX_RESPOSTAS_CN	Vetor com as respostas da parte objetiva da prova de Ciências da Natureza	Alfanumérico
TX_RESPOSTAS_CH	Vetor com as respostas da parte objetiva da prova de Ciências Humanas	Alfanumérico
TX_RESPOSTAS_LC	Vetor com as respostas da parte objetiva da prova de Linguagens e Códigos	Alfanumérico
TX_RESPOSTAS_MT	Vetor com as respostas da parte objetiva da prova de	Alfanumérico

	Matemática	
TP_LINGUA	Língua Estrangeira	Categórico
TX_GABARITO_CN	Vetor com o gabarito da parte objetiva da prova de Ciências da Natureza	Alfanumérico
TX_GABARITO_CH	Vetor com o gabarito da parte objetiva da prova de Ciências Humanas	Alfanumérico
TX_GABARITO_LC	Vetor com o gabarito da parte objetiva da prova de Linguagens e Códigos	Alfanumérico
TX_GABARITO_MT	Vetor com o gabarito da parte objetiva da prova de Matemática	Alfanumérico
Dados da Redação		
TP_STATUS_REDACAO	Situação da redação do participante	Categórico
NU_NOTA_COMP1	Nota da competência 1 - Demonstrar domínio da modalidade escrita formal da Língua Portuguesa	Numérico
NU_NOTA_COMP2	Nota da competência 2 - Compreender a proposta de redação e aplicar conceitos das várias áreas de conhecimento para desenvolver o tema, dentro dos limites estruturais do texto dissertativo-argumentativo em prosa	Numérico
NU_NOTA_COMP3	Nota da competência 3 - Selecionar, relacionar, organizar e interpretar informações, fatos, opiniões e argumentos em defesa de um ponto de vista	Numérico
NU_NOTA_COMP4	Nota da competência 4 - Demonstrar conhecimento dos mecanismos linguísticos necessários para a construção da argumentação	Numérico
NU_NOTA_COMP5	Nota da competência 5 - Elaborar proposta de intervenção para o problema abordado, respeitando os direitos humanos	Numérico
NU_NOTA_REDACAO	Nota da prova de redação	Numérico

Dados do Questionário Socioeconômico		
Q001	Até que série seu pai, ou o homem responsável por você, estudou?	Categórico
Q002	Até que série sua mãe, ou a mulher responsável por você, estudou?	Categórico
Q003	A partir da apresentação de algumas ocupações divididas em grupos ordenados, indique o grupo que contempla a ocupação mais próxima da ocupação do seu pai ou do homem responsável por você. (Se ele não estiver trabalhando, escolha uma ocupação pensando no último trabalho dele).	Categórico
Q004	A partir da apresentação de algumas ocupações divididas em grupos ordenados, indique o grupo que contempla a ocupação mais próxima da ocupação da sua mãe ou da mulher responsável por você. (Se ela não estiver trabalhando, escolha uma ocupação pensando no último trabalho dela).	Categórico
Q005	Incluindo você, quantas pessoas moram atualmente em sua residência?	Categórico
Q006	Qual é a renda mensal de sua família? (Some a sua renda com a dos seus familiares.)	Categórico
Q007	Em sua residência trabalha empregado(a) doméstico(a)?	Categórico
Q008	Na sua residência tem banheiro?	Categórico
Q009	Na sua residência tem quartos para dormir?	Categórico
Q010	Na sua residência tem carro?	Categórico
Q011	Na sua residência tem motocicleta?	Categórico
Q012	Na sua residência tem	Categórico

	geladeira?	
Q013	Na sua residência tem freezer (independente ou segunda porta da geladeira)?	Categórico
Q014	Na sua residência tem máquina de lavar roupa? (o tanquinho NÃO deve ser considerado)	Categórico
Q015	Na sua residência tem máquina de secar roupa (independente ou em conjunto com a máquina de lavar roupa)?	Categórico
Q016	Na sua residência tem forno micro-ondas?	Categórico
Q017	Na sua residência tem máquina de lavar louça?	Categórico
Q018	Na sua residência tem aspirador de pó?	Categórico
Q019	Na sua residência tem televisão em cores?	Categórico
Q020	Na sua residência tem aparelho de DVD?	Categórico
Q021	Na sua residência tem TV por assinatura?	Categórico
Q022	Na sua residência tem telefone celular?	Categórico
Q023	Na sua residência tem telefone fixo?	Categórico
Q024	Na sua residência tem computador?	Categórico
Q025	Na sua residência tem acesso à Internet?	Categórico
Q026	Você já concluiu ou está concluindo o Ensino Médio?	Categórico
Q027	Em que tipo de escola você frequentou o Ensino Médio?	Categórico

APÊNDICE A – LIMPEZA E PRÉ-PROCESSAMENTO DOS DADOS

Figura 7 – Seleção dos Atributos a Serem Utilizados

```
import pandas as pd

campos = ['NU_ANO', 'CO_MUNICIPIO_RESIDENCIA', 'CO_UF_RESIDENCIA', 'NU_IDADE', 'TP_SEXO',
          'TP_COR_RACA',  'NU_NOTA_CN', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_MT',
          'NU_NOTA_REDACAO', 'Q001', 'Q002', 'Q003', 'Q004', 'Q005', 'Q006', 'Q007',
          'Q008', 'Q009', 'Q010', 'Q011', 'Q012', 'Q014', 'Q015', 'Q016', 'Q017',
          'Q019', 'Q020', 'Q021', 'Q022', 'Q023', 'Q024', 'Q025', 'Q046', 'Q047']

dadosEnem = pd.read_csv('microdados_enem_2016.csv', sep=';', dtype=object,
                        encoding='ISO-8859-1', usecols=campos)

dadosEnem.to_csv('pos_etapa1_2016.csv', sep = ';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

Figura 8 – Remoção dos Registros Incompletos

```
import pandas as pd

dadosEnem = pd.read_csv('pos_etapa1_2016.csv', sep=';', dtype=object, encoding='utf-8')

dadosEnem = dadosEnem[dadosEnem.CO_UF_RESIDENCIA == '52']

dadosEnem = dadosEnem.dropna(subset=['NU_NOTA_CN', 'NU_NOTA_CH', 'NU_NOTA_LC',
                                     'NU_NOTA_MT', 'NU_NOTA_REDACAO'], axis=0)

dadosEnem = dadosEnem.dropna(axis=0)

dadosEnem.to_csv('pos_etapa2_2016.csv', sep=';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

APÊNDICE B – REDUÇÃO E TRANSFORMAÇÃO DOS DADOS

Figura 9 – Unificação das Bases de Dados

```
import pandas as pd

dadosEnem2016 = pd.read_csv('pos_etapa2_2016.csv', sep=';', dtype=object, encoding='utf-8')
dadosEnem2017 = pd.read_csv('pos_etapa2_2017.csv', sep=';', dtype=object, encoding='utf-8')
dadosEnem2018 = pd.read_csv('pos_etapa2_2018.csv', sep=';', dtype=object, encoding='utf-8')

dadosEnemFinal = pd.concat([dadosEnem2016, dadosEnem2017, dadosEnem2018])

dadosEnemFinal.to_csv('pos_etapa3_enem.csv', sep=';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

Figura 10 – Criação da Média das Notas das Provas Objetivas

```
import pandas as pd

campos = ['NU_NOTA_CN', 'NU_NOTA_CH', 'NU_NOTA_LC', 'NU_NOTA_MT']

notas = pd.read_csv('pos_etapa3_enem.csv', sep=';', encoding='utf-8', usecols=campos)

dadosEnem = pd.read_csv('pos_etapa3_enem.csv', sep=';', dtype=object, encoding='utf-8')

media = notas.mean(axis=1)
media = round(media, 1)

dadosEnem.insert(loc=9, column='MED_SEM_RED', value=media)

dadosEnem.to_csv('pos_etapa4_enem.csv', sep=';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

Figura 11 – Categorização dos Atributos Numéricos

```
import pandas as pd

dadosEnem = pd.read_csv('pos_etapa4_enem.csv', sep=';', encoding='utf-8')

#categorização dos municipios
municipiosRegMetro = [5200050, 5201405, 5201801, 5203302, 5203559, 5203609, 5204557,
                      5205208, 5208400, 5208806, 5209200, 5209705, 5210000, 5214507,
                      5215009, 5219738, 5220454, 5221197, 5221403, 5208707]
dadosEnem.loc[dadosEnem['CO_MUNICIPIO_RESIDENCIA'] == 5208707, 'CO_MUNICIPIO_RESIDENCIA'] = 'mun_capital'
for x in range(19):
    dadosEnem.loc[dadosEnem['CO_MUNICIPIO_RESIDENCIA'] == municipiosRegMetro[x],
                  'CO_MUNICIPIO_RESIDENCIA'] = 'mun_reg_metro'
dadosEnem.loc[(dadosEnem['CO_MUNICIPIO_RESIDENCIA'] != 'mun_capital') &
              (dadosEnem['CO_MUNICIPIO_RESIDENCIA'] != 'mun_reg_metro'),
                  'CO_MUNICIPIO_RESIDENCIA'] = 'mun_interior'
#categorização da idade
dadosEnem.loc[dadosEnem['NU_IDADE'] < 21, 'NU_IDADE'] = 0
dadosEnem.loc[(dadosEnem['NU_IDADE'] >= 21) & (dadosEnem['NU_IDADE'] < 31), 'NU_IDADE'] = 1
dadosEnem.loc[(dadosEnem['NU_IDADE'] >= 31) & (dadosEnem['NU_IDADE'] < 41), 'NU_IDADE'] = 2
dadosEnem.loc[dadosEnem['NU_IDADE'] >= 41, 'NU_IDADE'] = 3
dadosEnem.loc[dadosEnem['NU_IDADE'] == 0, 'NU_IDADE'] = 'idade_menor_21'
dadosEnem.loc[dadosEnem['NU_IDADE'] == 1, 'NU_IDADE'] = 'idade_21_30'
dadosEnem.loc[dadosEnem['NU_IDADE'] == 2, 'NU_IDADE'] = 'idade_31_40'
dadosEnem.loc[dadosEnem['NU_IDADE'] == 3, 'NU_IDADE'] = 'idade_maior_41'
#categorização das notas da provas objetivas
dadosEnem.loc[dadosEnem['MED_SEM_RED'] < 450, 'MED_SEM_RED'] = 0
dadosEnem.loc[(dadosEnem['MED_SEM_RED'] >= 450) & (dadosEnem['MED_SEM_RED'] < 550), 'MED_SEM_RED'] = 1
dadosEnem.loc[(dadosEnem['MED_SEM_RED'] >= 550) & (dadosEnem['MED_SEM_RED'] < 650), 'MED_SEM_RED'] = 2
dadosEnem.loc[(dadosEnem['MED_SEM_RED'] >= 650) & (dadosEnem['MED_SEM_RED'] < 750), 'MED_SEM_RED'] = 3
dadosEnem.loc[dadosEnem['MED_SEM_RED'] >= 750, 'MED_SEM_RED'] = 4
dadosEnem.loc[dadosEnem['MED_SEM_RED'] == 0, 'MED_SEM_RED'] = 'nota_ob_mt_baixa'
dadosEnem.loc[dadosEnem['MED_SEM_RED'] == 1, 'MED_SEM_RED'] = 'nota_ob_baixa'
dadosEnem.loc[dadosEnem['MED_SEM_RED'] == 2, 'MED_SEM_RED'] = 'nota_ob_regular'
dadosEnem.loc[dadosEnem['MED_SEM_RED'] == 3, 'MED_SEM_RED'] = 'nota_ob_alta'
dadosEnem.loc[dadosEnem['MED_SEM_RED'] == 4, 'MED_SEM_RED'] = 'nota_ob_mt_alta'
#categorização das notas da redação
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] < 500, 'NU_NOTA_REDACAO'] = 0
dadosEnem.loc[(dadosEnem['NU_NOTA_REDACAO'] >= 500) & (dadosEnem['NU_NOTA_REDACAO'] < 600), 'NU_NOTA_REDACAO'] = 1
dadosEnem.loc[(dadosEnem['NU_NOTA_REDACAO'] >= 600) & (dadosEnem['NU_NOTA_REDACAO'] < 700), 'NU_NOTA_REDACAO'] = 2
dadosEnem.loc[(dadosEnem['NU_NOTA_REDACAO'] >= 700) & (dadosEnem['NU_NOTA_REDACAO'] < 800), 'NU_NOTA_REDACAO'] = 3
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] >= 800, 'NU_NOTA_REDACAO'] = 4
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] == 0, 'NU_NOTA_REDACAO'] = 'nota_red_mt_baixa'
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] == 1, 'NU_NOTA_REDACAO'] = 'nota_red_baixa'
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] == 2, 'NU_NOTA_REDACAO'] = 'nota_red_regular'
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] == 3, 'NU_NOTA_REDACAO'] = 'nota_red_alta'
dadosEnem.loc[dadosEnem['NU_NOTA_REDACAO'] == 4, 'NU_NOTA_REDACAO'] = 'nota_red_mt_alta'
```

Fonte: Araújo e Silva (2020)

Figura 12 – Renomeação das Categorias 1

```
#categorização do ano
dadosEnem.loc[dadosEnem['NU_ANO'] == 2016, 'NU_ANO'] = 'ano_2016'
dadosEnem.loc[dadosEnem['NU_ANO'] == 2017, 'NU_ANO'] = 'ano_2017'
dadosEnem.loc[dadosEnem['NU_ANO'] == 2018, 'NU_ANO'] = 'ano_2018'
#categorização do sexo
dadosEnem.loc[dadosEnem['TP_SEXO'] == 'F', 'TP_SEXO'] = 'sexo_fem'
dadosEnem.loc[dadosEnem['TP_SEXO'] == 'M', 'TP_SEXO'] = 'sexo_mas'
#categorização da cor
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 0, 'TP_COR_RACA'] = 'cor_n_declarada'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 1, 'TP_COR_RACA'] = 'cor_branca'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 2, 'TP_COR_RACA'] = 'cor_preta'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 3, 'TP_COR_RACA'] = 'cor_parda'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 4, 'TP_COR_RACA'] = 'cor_amarela'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 5, 'TP_COR_RACA'] = 'cor_indigena'
dadosEnem.loc[dadosEnem['TP_COR_RACA'] == 6, 'TP_COR_RACA'] = 'cor_n_declarada'
#categorização da Q001
dadosEnem.loc[dadosEnem['Q001'] == 'A', 'Q001'] = 'esc_pai_sem_estudo'
dadosEnem.loc[dadosEnem['Q001'] == 'B', 'Q001'] = 'esc_pai_ens_fund_inc'
dadosEnem.loc[dadosEnem['Q001'] == 'C', 'Q001'] = 'esc_pai_ens_fund_inc'
dadosEnem.loc[dadosEnem['Q001'] == 'D', 'Q001'] = 'esc_pai_ens_fund_comp'
dadosEnem.loc[dadosEnem['Q001'] == 'E', 'Q001'] = 'esc_pai_ens_med_comp'
dadosEnem.loc[dadosEnem['Q001'] == 'F', 'Q001'] = 'esc_pai_ens_sup_comp'
dadosEnem.loc[dadosEnem['Q001'] == 'G', 'Q001'] = 'esc_pai_ens_sup_comp'
dadosEnem.loc[dadosEnem['Q001'] == 'H', 'Q001'] = 'esc_pai_nao_info'
#categorização da Q002
dadosEnem.loc[dadosEnem['Q002'] == 'A', 'Q002'] = 'esc_mae_sem_estudo'
dadosEnem.loc[dadosEnem['Q002'] == 'B', 'Q002'] = 'esc_mae_ens_fund_inc'
dadosEnem.loc[dadosEnem['Q002'] == 'C', 'Q002'] = 'esc_mae_ens_fund_inc'
dadosEnem.loc[dadosEnem['Q002'] == 'D', 'Q002'] = 'esc_mae_ens_fund_comp'
dadosEnem.loc[dadosEnem['Q002'] == 'E', 'Q002'] = 'esc_mae_ens_med_comp'
dadosEnem.loc[dadosEnem['Q002'] == 'F', 'Q002'] = 'esc_mae_ens_sup_comp'
dadosEnem.loc[dadosEnem['Q002'] == 'G', 'Q002'] = 'esc_mae_ens_sup_comp'
dadosEnem.loc[dadosEnem['Q002'] == 'H', 'Q002'] = 'esc_mae_nao_info'
#categorização da Q003
dadosEnem.loc[dadosEnem['Q003'] == 'A', 'Q003'] = 'prof_pai_a'
dadosEnem.loc[dadosEnem['Q003'] == 'B', 'Q003'] = 'prof_pai_b'
dadosEnem.loc[dadosEnem['Q003'] == 'C', 'Q003'] = 'prof_pai_c'
dadosEnem.loc[dadosEnem['Q003'] == 'D', 'Q003'] = 'prof_pai_d'
dadosEnem.loc[dadosEnem['Q003'] == 'E', 'Q003'] = 'prof_pai_e'
dadosEnem.loc[dadosEnem['Q003'] == 'F', 'Q003'] = 'prof_pai_nao_info'
#categorização da Q004
dadosEnem.loc[dadosEnem['Q004'] == 'A', 'Q004'] = 'prof_mae_a'
dadosEnem.loc[dadosEnem['Q004'] == 'B', 'Q004'] = 'prof_mae_b'
dadosEnem.loc[dadosEnem['Q004'] == 'C', 'Q004'] = 'prof_mae_c'
dadosEnem.loc[dadosEnem['Q004'] == 'D', 'Q004'] = 'prof_mae_d'
dadosEnem.loc[dadosEnem['Q004'] == 'E', 'Q004'] = 'prof_mae_e'
dadosEnem.loc[dadosEnem['Q004'] == 'F', 'Q004'] = 'prof_mae_nao_info'
#categorização da Q005
dadosEnem.loc[dadosEnem['Q005'] == 1, 'Q005'] = 1
dadosEnem.loc[(dadosEnem['Q005'] >= 2) & (dadosEnem['Q005'] < 5), 'Q005'] = 2
dadosEnem.loc[(dadosEnem['Q005'] >= 5) & (dadosEnem['Q005'] < 8), 'Q005'] = 3
dadosEnem.loc[(dadosEnem['Q005'] >= 8) & (dadosEnem['Q005'] < 11), 'Q005'] = 4
dadosEnem.loc[dadosEnem['Q005'] >= 11, 'Q005'] = 5
dadosEnem.loc[dadosEnem['Q005'] == 1, 'Q005'] = 'mora_sozinho'
dadosEnem.loc[dadosEnem['Q005'] == 2, 'Q005'] = 'mora_2_a_4'
dadosEnem.loc[dadosEnem['Q005'] == 3, 'Q005'] = 'mora_5_a_7'
dadosEnem.loc[dadosEnem['Q005'] == 4, 'Q005'] = 'mora_8_a_11'
dadosEnem.loc[dadosEnem['Q005'] == 5, 'Q005'] = 'mora_mais_11'
```

Fonte: Araújo e Silva (2020)

Figura 13 – Renomeação das Categorias 2

```
#categorização da Q006
dadosEnem.loc[dadosEnem['Q006'] == 'A', 'Q006'] = 'renda_ate_2_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'B', 'Q006'] = 'renda_ate_2_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'C', 'Q006'] = 'renda_ate_2_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'D', 'Q006'] = 'renda_ate_2_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'E', 'Q006'] = 'renda_2_ate_4_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'F', 'Q006'] = 'renda_2_ate_4_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'G', 'Q006'] = 'renda_2_ate_4_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'H', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'I', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'J', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'K', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'L', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'M', 'Q006'] = 'renda_4_ate_10_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'N', 'Q006'] = 'renda_10_ate_20_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'O', 'Q006'] = 'renda_10_ate_20_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'P', 'Q006'] = 'renda_10_ate_20_sal'
dadosEnem.loc[dadosEnem['Q006'] == 'Q', 'Q006'] = 'renda_mais_20'
#categorização da Q007
dadosEnem.loc[dadosEnem['Q007'] == 'A', 'Q007'] = 'empreg_dom_nao'
dadosEnem.loc[dadosEnem['Q007'] == 'B', 'Q007'] = 'empreg_dom_sim'
dadosEnem.loc[dadosEnem['Q007'] == 'C', 'Q007'] = 'empreg_dom_sim'
dadosEnem.loc[dadosEnem['Q007'] == 'D', 'Q007'] = 'empreg_dom_sim'
#categorização da Q008
dadosEnem.loc[dadosEnem['Q008'] == 'A', 'Q008'] = 'banheiro_nao'
dadosEnem.loc[dadosEnem['Q008'] == 'B', 'Q008'] = 'banheiro_um'
dadosEnem.loc[dadosEnem['Q008'] == 'C', 'Q008'] = 'banheiro_dois'
dadosEnem.loc[dadosEnem['Q008'] == 'D', 'Q008'] = 'banheiro_tres_mais'
dadosEnem.loc[dadosEnem['Q008'] == 'E', 'Q008'] = 'banheiro_tres_mais'
#categorização da Q009
dadosEnem.loc[dadosEnem['Q009'] == 'A', 'Q009'] = 'quartos_nao'
dadosEnem.loc[dadosEnem['Q009'] == 'B', 'Q009'] = 'quartos_um'
dadosEnem.loc[dadosEnem['Q009'] == 'C', 'Q009'] = 'quartos_dois'
dadosEnem.loc[dadosEnem['Q009'] == 'D', 'Q009'] = 'quartos_tres_mais'
dadosEnem.loc[dadosEnem['Q009'] == 'E', 'Q009'] = 'quartos_tres_mais'
#categorização da Q010
dadosEnem.loc[dadosEnem['Q010'] == 'A', 'Q010'] = 'carro_nao'
dadosEnem.loc[dadosEnem['Q010'] == 'B', 'Q010'] = 'carro_um'
dadosEnem.loc[dadosEnem['Q010'] == 'C', 'Q010'] = 'carro_dois'
dadosEnem.loc[dadosEnem['Q010'] == 'D', 'Q010'] = 'carro_tres_mais'
dadosEnem.loc[dadosEnem['Q010'] == 'E', 'Q010'] = 'carro_tres_mais'
#categorização da Q011
dadosEnem.loc[dadosEnem['Q011'] == 'A', 'Q011'] = 'moto_nao'
dadosEnem.loc[dadosEnem['Q011'] == 'B', 'Q011'] = 'moto_um'
dadosEnem.loc[dadosEnem['Q011'] == 'C', 'Q011'] = 'moto_dois'
dadosEnem.loc[dadosEnem['Q011'] == 'D', 'Q011'] = 'moto_tres_mais'
dadosEnem.loc[dadosEnem['Q011'] == 'E', 'Q011'] = 'moto_tres_mais'
#categorização da Q012
dadosEnem.loc[dadosEnem['Q012'] == 'A', 'Q012'] = 'gelad_nao'
dadosEnem.loc[dadosEnem['Q012'] == 'B', 'Q012'] = 'gelad_um'
dadosEnem.loc[dadosEnem['Q012'] == 'C', 'Q012'] = 'gelad_dois'
dadosEnem.loc[dadosEnem['Q012'] == 'D', 'Q012'] = 'gelad_tres_mais'
dadosEnem.loc[dadosEnem['Q012'] == 'E', 'Q012'] = 'gelad_tres_mais'
```

Fonte: Araújo e Silva (2020)

Figura 14 – Renomeação das Categorias 3

```
#categorização da Q014
dadosEnem.loc[dadosEnem['Q014'] == 'A', 'Q014'] = 'maq_lav_nao'
dadosEnem.loc[dadosEnem['Q014'] == 'B', 'Q014'] = 'maq_lav_um'
dadosEnem.loc[dadosEnem['Q014'] == 'C', 'Q014'] = 'maq_lav_dois'
dadosEnem.loc[dadosEnem['Q014'] == 'D', 'Q014'] = 'maq_lav_tres_mais'
dadosEnem.loc[dadosEnem['Q014'] == 'E', 'Q014'] = 'maq_lav_tres_mais'
#categorização da Q015
dadosEnem.loc[dadosEnem['Q015'] == 'A', 'Q015'] = 'maq_sec_nao'
dadosEnem.loc[dadosEnem['Q015'] == 'B', 'Q015'] = 'maq_sec_um'
dadosEnem.loc[dadosEnem['Q015'] == 'C', 'Q015'] = 'maq_sec_dois'
dadosEnem.loc[dadosEnem['Q015'] == 'D', 'Q015'] = 'maq_sec_tres_mais'
dadosEnem.loc[dadosEnem['Q015'] == 'E', 'Q015'] = 'maq_sec_tres_mais'
#categorização da Q016
dadosEnem.loc[dadosEnem['Q016'] == 'A', 'Q016'] = 'mic_ondas_nao'
dadosEnem.loc[dadosEnem['Q016'] == 'B', 'Q016'] = 'mic_ondas_um'
dadosEnem.loc[dadosEnem['Q016'] == 'C', 'Q016'] = 'mic_ondas_dois'
dadosEnem.loc[dadosEnem['Q016'] == 'D', 'Q016'] = 'mic_ondas_tres_mais'
dadosEnem.loc[dadosEnem['Q016'] == 'E', 'Q016'] = 'mic_ondas_tres_mais'
#categorização da Q017
dadosEnem.loc[dadosEnem['Q017'] == 'A', 'Q017'] = 'maq_lav_louca_nao'
dadosEnem.loc[dadosEnem['Q017'] == 'B', 'Q017'] = 'maq_lav_louca_um'
dadosEnem.loc[dadosEnem['Q017'] == 'C', 'Q017'] = 'maq_lav_louca_dois'
dadosEnem.loc[dadosEnem['Q017'] == 'D', 'Q017'] = 'maq_lav_louca_tres_mais'
dadosEnem.loc[dadosEnem['Q017'] == 'E', 'Q017'] = 'maq_lav_louca_tres_mais'
#categorização da Q019
dadosEnem.loc[dadosEnem['Q019'] == 'A', 'Q019'] = 'tv_nao'
dadosEnem.loc[dadosEnem['Q019'] == 'B', 'Q019'] = 'tv_um'
dadosEnem.loc[dadosEnem['Q019'] == 'C', 'Q019'] = 'tv_dois'
dadosEnem.loc[dadosEnem['Q019'] == 'D', 'Q019'] = 'tv_tres_mais'
dadosEnem.loc[dadosEnem['Q019'] == 'E', 'Q019'] = 'tv_tres_mais'
#categorização da Q020
dadosEnem.loc[dadosEnem['Q020'] == 'A', 'Q020'] = 'dvd_nao'
dadosEnem.loc[dadosEnem['Q020'] == 'B', 'Q020'] = 'dvd_sim'
#categorização da Q021
dadosEnem.loc[dadosEnem['Q021'] == 'A', 'Q021'] = 'tv_assin_nao'
dadosEnem.loc[dadosEnem['Q021'] == 'B', 'Q021'] = 'tv_assin_sim'
#categorização da Q022
dadosEnem.loc[dadosEnem['Q022'] == 'A', 'Q022'] = 'celular_nao'
dadosEnem.loc[dadosEnem['Q022'] == 'B', 'Q022'] = 'celular_um'
dadosEnem.loc[dadosEnem['Q022'] == 'C', 'Q022'] = 'celular_dois'
dadosEnem.loc[dadosEnem['Q022'] == 'D', 'Q022'] = 'celular_tres_mais'
dadosEnem.loc[dadosEnem['Q022'] == 'E', 'Q022'] = 'celular_tres_mais'
#categorização da Q023
dadosEnem.loc[dadosEnem['Q023'] == 'A', 'Q023'] = 'tel_fix_nao'
dadosEnem.loc[dadosEnem['Q023'] == 'B', 'Q023'] = 'tel_fix_sim'
#categorização da Q024
dadosEnem.loc[dadosEnem['Q024'] == 'A', 'Q024'] = 'comput_nao'
dadosEnem.loc[dadosEnem['Q024'] == 'B', 'Q024'] = 'comput_um'
dadosEnem.loc[dadosEnem['Q024'] == 'C', 'Q024'] = 'comput_dois'
dadosEnem.loc[dadosEnem['Q024'] == 'D', 'Q024'] = 'comput_tres_mais'
dadosEnem.loc[dadosEnem['Q024'] == 'E', 'Q024'] = 'comput_tres_mais'
#categorização da Q025
dadosEnem.loc[dadosEnem['Q025'] == 'A', 'Q025'] = 'internet_nao'
dadosEnem.loc[dadosEnem['Q025'] == 'B', 'Q025'] = 'internet_sim'
#categorização da Q026
dadosEnem.loc[dadosEnem['Q026'] == 'A', 'Q026'] = 'ens_med_concluido'
dadosEnem.loc[dadosEnem['Q026'] == 'B', 'Q026'] = 'ens_med_conc_ano_atual'
dadosEnem.loc[dadosEnem['Q026'] == 'C', 'Q026'] = 'ens_med_conc_prox_ano'
dadosEnem.loc[dadosEnem['Q026'] == 'D', 'Q026'] = 'ens_med_nao_curs'
#categorização da Q027
dadosEnem.loc[dadosEnem['Q027'] == 'A', 'Q027'] = 'ens_publica'
dadosEnem.loc[dadosEnem['Q027'] == 'B', 'Q027'] = 'ens_pub_priv_sem_bolsa'
dadosEnem.loc[dadosEnem['Q027'] == 'C', 'Q027'] = 'ens_pub_priv_com_bolsa'
dadosEnem.loc[dadosEnem['Q027'] == 'D', 'Q027'] = 'ens_priv_sem_bolsa'
dadosEnem.loc[dadosEnem['Q027'] == 'E', 'Q027'] = 'ens_priv_com_bolsa'
dadosEnem.loc[dadosEnem['Q027'] == 'F', 'Q027'] = 'ens_nao_estud'

dadosEnem.to_csv('pos_etapa5_enem.csv', sep=';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)

Figura 15 – Transformação das Categorias em Colunas 1

```
import pandas as pd
import numpy as np

dadosEnem = pd.read_csv('pos_etapa5_enem.csv', sep=';', encoding='utf-8')

enemBin = pd.DataFrame()

enemBin['ano_2016'] = np.where(dadosEnem['NU_ANO'] == 'ano_2016', '1', '0')
enemBin['ano_2017'] = np.where(dadosEnem['NU_ANO'] == 'ano_2017', '1', '0')
enemBin['ano_2018'] = np.where(dadosEnem['NU_ANO'] == 'ano_2018', '1', '0')

enemBin['mun_capital'] = np.where(dadosEnem['CO_MUNICIPIO_RESIDENCIA'] == 'mun_capital', '1', '0')
enemBin['mun_reg_metro'] = np.where(dadosEnem['CO_MUNICIPIO_RESIDENCIA'] == 'mun_reg_metro', '1', '0')
enemBin['mun_interior'] = np.where(dadosEnem['CO_MUNICIPIO_RESIDENCIA'] == 'mun_interior', '1', '0')

enemBin['idade_menor_21'] = np.where(dadosEnem['NU_IDADE'] == 'idade_menor_21', '1', '0')
enemBin['idade_21_30'] = np.where(dadosEnem['NU_IDADE'] == 'idade_21_30', '1', '0')
enemBin['idade_31_40'] = np.where(dadosEnem['NU_IDADE'] == 'idade_31_40', '1', '0')
enemBin['idade_maior_41'] = np.where(dadosEnem['NU_IDADE'] == 'idade_maior_41', '1', '0')

enemBin['sexo_fem'] = np.where(dadosEnem['TP_SEXO'] == 'sexo_fem', '1', '0')
enemBin['sexo_mas'] = np.where(dadosEnem['TP_SEXO'] == 'sexo_mas', '1', '0')

enemBin['cor_n_declarada'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_n_declarada', '1', '0')
enemBin['cor_branca'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_branca', '1', '0')
enemBin['cor_preta'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_preta', '1', '0')
enemBin['cor_parda'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_parda', '1', '0')
enemBin['cor_amarela'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_amarela', '1', '0')
enemBin['cor_indigena'] = np.where(dadosEnem['TP_COR_RACA'] == 'cor_indigena', '1', '0')

enemBin['nota_ob_mt_baixa'] = np.where(dadosEnem['MED_SEM_RED'] == 'nota_ob_mt_baixa', '1', '0')
enemBin['nota_ob_baixa'] = np.where(dadosEnem['MED_SEM_RED'] == 'nota_ob_baixa', '1', '0')
enemBin['nota_ob_regular'] = np.where(dadosEnem['MED_SEM_RED'] == 'nota_ob_regular', '1', '0')
enemBin['nota_ob_alta'] = np.where(dadosEnem['MED_SEM_RED'] == 'nota_ob_alta', '1', '0')
enemBin['nota_ob_mt_alta'] = np.where(dadosEnem['MED_SEM_RED'] == 'nota_ob_mt_alta', '1', '0')

enemBin['nota_red_mt_baixa'] = np.where(dadosEnem['NU_NOTA_REDACAO'] == 'nota_red_mt_baixa', '1', '0')
enemBin['nota_red_baixa'] = np.where(dadosEnem['NU_NOTA_REDACAO'] == 'nota_red_baixa', '1', '0')
enemBin['nota_red_regular'] = np.where(dadosEnem['NU_NOTA_REDACAO'] == 'nota_red_regular', '1', '0')
enemBin['nota_red_alta'] = np.where(dadosEnem['NU_NOTA_REDACAO'] == 'nota_red_alta', '1', '0')
enemBin['nota_red_mt_alta'] = np.where(dadosEnem['NU_NOTA_REDACAO'] == 'nota_red_mt_alta', '1', '0')

enemBin['esc_pai_sem_estudo'] = np.where(dadosEnem['Q001'] == 'esc_pai_sem_estudo', '1', '0')
enemBin['esc_pai_ens_fund_inc'] = np.where(dadosEnem['Q001'] == 'esc_pai_ens_fund_inc', '1', '0')
enemBin['esc_pai_ens_fund_comp'] = np.where(dadosEnem['Q001'] == 'esc_pai_ens_fund_comp', '1', '0')
enemBin['esc_pai_ens_med_comp'] = np.where(dadosEnem['Q001'] == 'esc_pai_ens_med_comp', '1', '0')
enemBin['esc_pai_ens_sup_comp'] = np.where(dadosEnem['Q001'] == 'esc_pai_ens_sup_comp', '1', '0')
enemBin['esc_pai_nao_info'] = np.where(dadosEnem['Q001'] == 'esc_pai_nao_info', '1', '0')

enemBin['esc_mae_sem_estudo'] = np.where(dadosEnem['Q002'] == 'esc_mae_sem_estudo', '1', '0')
enemBin['esc_mae_ens_fund_inc'] = np.where(dadosEnem['Q002'] == 'esc_mae_ens_fund_inc', '1', '0')
enemBin['esc_mae_ens_fund_comp'] = np.where(dadosEnem['Q002'] == 'esc_mae_ens_fund_comp', '1', '0')
enemBin['esc_mae_ens_med_comp'] = np.where(dadosEnem['Q002'] == 'esc_mae_ens_med_comp', '1', '0')
enemBin['esc_mae_ens_sup_comp'] = np.where(dadosEnem['Q002'] == 'esc_mae_ens_sup_comp', '1', '0')
enemBin['esc_mae_nao_info'] = np.where(dadosEnem['Q002'] == 'esc_mae_nao_info', '1', '0')

enemBin['prof_pai_a'] = np.where(dadosEnem['Q003'] == 'prof_pai_a', '1', '0')
enemBin['prof_pai_b'] = np.where(dadosEnem['Q003'] == 'prof_pai_b', '1', '0')
enemBin['prof_pai_c'] = np.where(dadosEnem['Q003'] == 'prof_pai_c', '1', '0')
enemBin['prof_pai_d'] = np.where(dadosEnem['Q003'] == 'prof_pai_d', '1', '0')
enemBin['prof_pai_e'] = np.where(dadosEnem['Q003'] == 'prof_pai_e', '1', '0')
enemBin['prof_pai_nao_info'] = np.where(dadosEnem['Q003'] == 'prof_pai_nao_info', '1', '0')
```

Fonte: Araújo e Silva (2020)

Figura 16 – Transformação das Categorias em Colunas 2

```
enemBin['prof_mae_a'] = np.where(dadosEnem['Q004'] == 'prof_mae_a', '1', '0')
enemBin['prof_mae_b'] = np.where(dadosEnem['Q004'] == 'prof_mae_b', '1', '0')
enemBin['prof_mae_c'] = np.where(dadosEnem['Q004'] == 'prof_mae_c', '1', '0')
enemBin['prof_mae_d'] = np.where(dadosEnem['Q004'] == 'prof_mae_d', '1', '0')
enemBin['prof_mae_e'] = np.where(dadosEnem['Q004'] == 'prof_mae_e', '1', '0')
enemBin['prof_mae_nao_info'] = np.where(dadosEnem['Q004'] == 'prof_mae_nao_info', '1', '0')

enemBin['mora_sozinho'] = np.where(dadosEnem['Q005'] == 'mora_sozinho', '1', '0')
enemBin['mora_2_a_4'] = np.where(dadosEnem['Q005'] == 'mora_2_a_4', '1', '0')
enemBin['mora_5_a_7'] = np.where(dadosEnem['Q005'] == 'mora_5_a_7', '1', '0')
enemBin['mora_8_a_11'] = np.where(dadosEnem['Q005'] == 'mora_8_a_11', '1', '0')
enemBin['mora_mais_11'] = np.where(dadosEnem['Q005'] == 'mora_mais_11', '1', '0')

enemBin['renda_ate_2_sal'] = np.where(dadosEnem['Q006'] == 'renda_ate_2_sal', '1', '0')
enemBin['renda_2_ate_4_sal'] = np.where(dadosEnem['Q006'] == 'renda_2_ate_4_sal', '1', '0')
enemBin['renda_4_ate_10_sal'] = np.where(dadosEnem['Q006'] == 'renda_4_ate_10_sal', '1', '0')
enemBin['renda_10_ate_20_sal'] = np.where(dadosEnem['Q006'] == 'renda_10_ate_20_sal', '1', '0')
enemBin['renda_mais_20_sal'] = np.where(dadosEnem['Q006'] == 'renda_mais_20_sal', '1', '0')

enemBin['empreg_dom_nao'] = np.where(dadosEnem['Q007'] == 'empreg_dom_nao', '1', '0')
enemBin['empreg_dom_sim'] = np.where(dadosEnem['Q007'] == 'empreg_dom_sim', '1', '0')

enemBin['banheiro_nao'] = np.where(dadosEnem['Q008'] == 'banheiro_nao', '1', '0')
enemBin['banheiro_um'] = np.where(dadosEnem['Q008'] == 'banheiro_um', '1', '0')
enemBin['banheiro_dois'] = np.where(dadosEnem['Q008'] == 'banheiro_dois', '1', '0')
enemBin['banheiro_tres_mais'] = np.where(dadosEnem['Q008'] == 'banheiro_tres_mais', '1', '0')

enemBin['quartos_nao'] = np.where(dadosEnem['Q009'] == 'quartos_nao', '1', '0')
enemBin['quartos_um'] = np.where(dadosEnem['Q009'] == 'quartos_um', '1', '0')
enemBin['quartos_dois'] = np.where(dadosEnem['Q009'] == 'quartos_dois', '1', '0')
enemBin['quartos_tres_mais'] = np.where(dadosEnem['Q009'] == 'quartos_tres_mais', '1', '0')

enemBin['carro_nao'] = np.where(dadosEnem['Q010'] == 'carro_nao', '1', '0')
enemBin['carro_um'] = np.where(dadosEnem['Q010'] == 'carro_um', '1', '0')
enemBin['carro_dois'] = np.where(dadosEnem['Q010'] == 'carro_dois', '1', '0')
enemBin['carro_tres_mais'] = np.where(dadosEnem['Q010'] == 'carro_tres_mais', '1', '0')

enemBin['moto_nao'] = np.where(dadosEnem['Q011'] == 'moto_nao', '1', '0')
enemBin['moto_um'] = np.where(dadosEnem['Q011'] == 'moto_um', '1', '0')
enemBin['moto_dois'] = np.where(dadosEnem['Q011'] == 'moto_dois', '1', '0')
enemBin['moto_tres_mais'] = np.where(dadosEnem['Q011'] == 'moto_tres_mais', '1', '0')

enemBin['gelad_nao'] = np.where(dadosEnem['Q012'] == 'gelad_nao', '1', '0')
enemBin['gelad_um'] = np.where(dadosEnem['Q012'] == 'gelad_um', '1', '0')
enemBin['gelad_dois'] = np.where(dadosEnem['Q012'] == 'gelad_dois', '1', '0')
enemBin['gelad_tres_mais'] = np.where(dadosEnem['Q012'] == 'gelad_tres_mais', '1', '0')

enemBin['maq_lav_nao'] = np.where(dadosEnem['Q014'] == 'maq_lav_nao', '1', '0')
enemBin['maq_lav_um'] = np.where(dadosEnem['Q014'] == 'maq_lav_um', '1', '0')
enemBin['maq_lav_dois'] = np.where(dadosEnem['Q014'] == 'maq_lav_dois', '1', '0')
enemBin['maq_lav_tres_mais'] = np.where(dadosEnem['Q014'] == 'maq_lav_tres_mais', '1', '0')

enemBin['maq_sec_nao'] = np.where(dadosEnem['Q015'] == 'maq_sec_nao', '1', '0')
enemBin['maq_sec_um'] = np.where(dadosEnem['Q015'] == 'maq_sec_um', '1', '0')
enemBin['maq_sec_dois'] = np.where(dadosEnem['Q015'] == 'maq_sec_dois', '1', '0')
enemBin['maq_sec_tres_mais'] = np.where(dadosEnem['Q015'] == 'maq_sec_tres_mais', '1', '0')
```

Fonte: Araújo e Silva (2020)

Figura 17 – Transformação das Categorias em Colunas 3

```
enemBin['mic_ondas_nao'] = np.where(dadosEnem['Q016'] == 'mic_ondas_nao', '1', '0')
enemBin['mic_ondas_um'] = np.where(dadosEnem['Q016'] == 'mic_ondas_um', '1', '0')
enemBin['mic_ondas_dois'] = np.where(dadosEnem['Q016'] == 'mic_ondas_dois', '1', '0')
enemBin['mic_ondas_tres_mais'] = np.where(dadosEnem['Q016'] == 'mic_ondas_tres_mais', '1', '0')

enemBin['maq_lav_louca_nao'] = np.where(dadosEnem['Q017'] == 'maq_lav_louca_nao', '1', '0')
enemBin['maq_lav_louca_um'] = np.where(dadosEnem['Q017'] == 'maq_lav_louca_um', '1', '0')
enemBin['maq_lav_louca_dois'] = np.where(dadosEnem['Q017'] == 'maq_lav_louca_dois', '1', '0')
enemBin['maq_lav_louca_tres_mais'] = np.where(dadosEnem['Q017'] == 'maq_lav_louca_tres_mais', '1', '0')

enemBin['tv_nao'] = np.where(dadosEnem['Q019'] == 'tv_nao', '1', '0')
enemBin['tv_um'] = np.where(dadosEnem['Q019'] == 'tv_um', '1', '0')
enemBin['tv_dois'] = np.where(dadosEnem['Q019'] == 'tv_dois', '1', '0')
enemBin['tv_tres_mais'] = np.where(dadosEnem['Q019'] == 'tv_tres_mais', '1', '0')

enemBin['dvd_nao'] = np.where(dadosEnem['Q020'] == 'dvd_nao', '1', '0')
enemBin['dvd_sim'] = np.where(dadosEnem['Q020'] == 'dvd_sim', '1', '0')

enemBin['tv_assin_nao'] = np.where(dadosEnem['Q021'] == 'tv_assin_nao', '1', '0')
enemBin['tv_assin_sim'] = np.where(dadosEnem['Q021'] == 'tv_assin_sim', '1', '0')

enemBin['celular_nao'] = np.where(dadosEnem['Q022'] == 'celular_nao', '1', '0')
enemBin['celular_um'] = np.where(dadosEnem['Q022'] == 'celular_um', '1', '0')
enemBin['celular_dois'] = np.where(dadosEnem['Q022'] == 'celular_dois', '1', '0')
enemBin['celular_tres_mais'] = np.where(dadosEnem['Q022'] == 'celular_tres_mais', '1', '0')

enemBin['tel_fix_nao'] = np.where(dadosEnem['Q023'] == 'tel_fix_nao', '1', '0')
enemBin['tel_fix_sim'] = np.where(dadosEnem['Q023'] == 'tel_fix_sim', '1', '0')

enemBin['comput_nao'] = np.where(dadosEnem['Q024'] == 'comput_nao', '1', '0')
enemBin['comput_um'] = np.where(dadosEnem['Q024'] == 'comput_um', '1', '0')
enemBin['comput_dois'] = np.where(dadosEnem['Q024'] == 'comput_dois', '1', '0')
enemBin['comput_tres_mais'] = np.where(dadosEnem['Q024'] == 'comput_tres_mais', '1', '0')

enemBin['internet_nao'] = np.where(dadosEnem['Q025'] == 'internet_nao', '1', '0')
enemBin['internet_sim'] = np.where(dadosEnem['Q025'] == 'internet_sim', '1', '0')

enemBin['ens_med_concluido'] = np.where(dadosEnem['Q026'] == 'ens_med_concluido', '1', '0')
enemBin['ens_med_conc_ano_atual'] = np.where(dadosEnem['Q026'] == 'ens_med_conc_ano_atual', '1', '0')
enemBin['ens_med_conc_prox_ano'] = np.where(dadosEnem['Q026'] == 'ens_med_conc_prox_ano', '1', '0')
enemBin['ens_med_nao_curs'] = np.where(dadosEnem['Q026'] == 'ens_med_nao_curs', '1', '0')

enemBin['ens_publica'] = np.where(dadosEnem['Q027'] == 'ens_publica', '1', '0')
enemBin['ens_pub_priv_sem_bolsa'] = np.where(dadosEnem['Q027'] == 'ens_pub_priv_sem_bolsa', '1', '0')
enemBin['ens_pub_priv_com_bolsa'] = np.where(dadosEnem['Q027'] == 'ens_pub_priv_com_bolsa', '1', '0')
enemBin['ens_priv_sem_bolsa'] = np.where(dadosEnem['Q027'] == 'ens_priv_sem_bolsa', '1', '0')
enemBin['ens_priv_com_bolsa'] = np.where(dadosEnem['Q027'] == 'ens_priv_com_bolsa', '1', '0')
enemBin['ens_nao_estud'] = np.where(dadosEnem['Q027'] == 'ens_nao_estud', '1', '0')

enemBin.to_csv('pos_etapa6_enem.csv', sep=';', index=False, encoding='utf-8')
```

Fonte: Araújo e Silva (2020)