CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Renato Marinho Alves

Plataforma de visualização dos dados minerados dos dados do ENADE da área de Computação dos anos de 2008 a 2014

Palmas – TO

2018

Renato Marinho Alves

Plataforma de visualização dos dados minerados dos dados do ENADE da área de Computação dos anos de 2008 a 2014

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. M.e Heloise Acco Tives Leão.

Palmas – TO

2018

Renato Marinho Alves

Plataforma de mineração e apresentação dos dados do ENADE da área de Computação dos anos de 2008 a 2014

Trabalho de Conclusão de Curso (TCC) II elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Profa. M.e Heloise Acco Tives Leão.

Aprovado em: _____/_____/_____

BANCA EXAMINADORA

_____

Profa. M.e Heloise Acco Tives Leão

Orientador

Centro Universitário Luterano de Palmas – CEULP

_____

Prof. M. Fabiano Fagundes

Centro Universitário Luterano de Palmas

_____

Prof. D.ra Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas

Palmas – TO

2018

Dedico este trabalho a Deus e a minha família por terem me auxiliado em minhas escolhas e decisões em cada passo desta caminhada.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me dado forças e sabedoria para continuar lutando dia após dia para alcançar o objetivo de conseguir a graduação.

A minha família que sempre esteve me apoiando e me animando em toda a caminhada para a conclusão desta etapa.

Ao Corpo Docente do CEULP/ULBRA por sempre darem a direção e auxiliarem a partir de conselhos e conversas.

A minha professora e orientadora Profa. M.e. Heloíse Acco Tives Leão, que sempre me auxiliou desde a escrita até mesmo na implementação do projeto, mesmo não sendo sua especialidade, respondendo os e-mails de dúvidas com impressionante velocidade.

Agradeço ao Prof. M.Sc. Jackson de Sousa Gomes por ter me aceitado na Fábrica de Software, por ter sido meu chefe e por ter se disposto a me orientar em diversas situações passando conhecimentos que jamais pensei que iria adquirir.

Agradeço ao Prof. M.Sc. Fabiano Fagundes por me orientar nos projetos de Iniciação Científica, pela disponibilidade para tirar dúvidas e conversas de incentivo.

A todos que participaram direta ou indiretamente da minha formação, deixo a vocês meus sinceros agradecimentos.

RESUMO

ALVES, Renato Marinho. **Plataforma de mineração e apresentação dos dados do ENADE da área de Computação dos anos de 2008 a 2014.** 2018. 75 f. Trabalho de Conclusão de Curso (Graduação) - Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2018.

A procura crescente por informações relevantes extraídas de grandes volumes de dados vem tornando o emprego de tecnologias computacionais essencial para a obtenção de novos conhecimentos. Nesse contexto surge o conceito computacional de mineração de dados para auxiliar nesta tarefa. Sua evolução pode ser verificada pela ampla variedade de algoritmos existentes atualmente para atender diferentes necessidades, assim como pela criação de metodologias para otimizar seus processos. Bases de dados abertos são fontes valiosas de informações latentes e pouco exploradas. Utilizando esses recursos, este trabalho objetivou a criação de uma plataforma para visualização dos resultados da mineração dos microdados do ENADE, disponibilizados pelo INEP, de modo a obter-se novas informações sobre os acadêmicos da área de Computação nos anos 2005, 2008, 2011 e 2014. Para isso, foram aplicados algoritmos de classificação, clusterização e associação de forma a obter resultados passíveis de serem analisados, validados e que pudessem ser disponibilizados na plataforma por meio gráfico. O desenvolvimento do trabalho segue os passos estabelecidos no modelo de referência CRISP-DM, com algumas adaptações necessárias para atender aos objetivos do trabalho. Como resultado deste trabalho foi disponibilizada a plataforma Enade-DM de forma pública que permite a visualização dos resultados obtidos da aplicação dos algoritmos de mineração de dados sobre as respostas do ENADE de forma dinâmica por meio de filtros e gráficos.

Palavras-chave: Mineração de dados. CRISP-DM. Enade. Algoritmos de mineração de dados.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CC - Ciência da Computação

CRISP-DM - Cross Industry Standard Process for Data Mining

MD - Mineração de dados

BD - Banco de dados

ENADE - Exame Nacional de Desempenho de Estudantes

MinConf - Confiança mínima

MinSup - Suporte mínimo

SI - Sistemas de Informação

DW - Data Warehouse

Front-end - Parte visual de uma aplicação, programa ou website, bem como a coleta de informações a serem passadas à API e Back-end

Back-end - Parte que abrange a manipulação de dados do banco e tratamento de informações para consultas de acordo com o passado pelo Front-end

ETL - Extract Transform Load, em português Extração, Transformação e Carga

OLAP - Online Analytical Processing, em português Processamento Analítico Online

API - Interface de Programação de Aplicações

PDI - Pentaho Data Integration

SUMÁRIO

1 INTRODUÇÃO

Atualmente no Brasil tem-se o Exame Nacional de Desempenho de Estudantes (ENADE) como forma de mensurar a qualidade de ensino de instituições de ensino superior para cada curso que possuem. Segundo o Inep (2008) o "ENADE avalia o rendimento de concluintes dos cursos de graduação, em relação aos conteúdos programáticos, habilidade e competências adquiridas em sua formação". Seu objetivo é ponderar o desempenho dos estudantes em função dos conteúdos programáticos previstos na grade curricular dos cursos de graduação. Os dados referentes ao ENADE são disponibilizados de forma pública e gratuita pelo portal do Inep.

A mineração de dados (MD) surgiu da necessidade de se extrair informações de grandes bases de dados de maneira rápida e eficaz, tendo em vista que o processo manual demanda muito tempo. Nela utilizam-se algoritmos computacionais sobre os dados a fim de se descobrir novos padrões que tragam informações relevantes. Segundo Olson e Delen (2008) para que a mineração de dados seja feita, é necessária a identificação de um problema, que juntamente a uma coleção de dados pode conduzir a um mais amplo entendimento, e um modelo computacional capaz de prover análises estatísticas.

Atualmente é possível observar a busca constante de informações que possam auxiliar no desenvolvimento de um negócio ou pesquisa. Neste âmbito encontram-se os processos de mineração e análise de dados, de modo a se identificarem padrões e adquirir informações sobre os dados em que o processo está sendo aplicado.

Este trabalho propõe a utilização de técnicas de MD para a obtenção e análise das respostas dos discentes da área de Computação no ENADE por meio da seguinte questão: em quais campos de estudo da Computação que os discentes dos cursos da Computação possuem melhor desempenho de acordo com seu curso.

Para atingir a proposta do trabalho é abordado o uso de técnicas de extração, transformação, mineração e análise dos dados fornecidos pelo INEP sobre as respostas ao ENADE, com foco os dados da área de Computação dos anos de 2008, 2011 e 2014. A análise de informações se restringe aos cursos da área de Computação para que sejam analisadas as respostas dos discentes dos cursos desta área e estabelecer sua relação com os campos de estudos da Computação. Desta forma este trabalho objetiva a análise dos dados do ENADE a fim de obter informações sobre o desempenho dos discentes da área de Computação, de modo a observar seus desempenhos nos diversos campos de estudo da área em questão.

De modo a alcançar os objetivos do trabalho foi utilizado como base o modelo de referência CRISP-DM, de maneira a otimizar o processo e obtenção de resultados da aplicação das técnicas. Os detalhes do desenvolvimento da plataforma de visualização dos dados minerados serão descritos de forma a esclarecer as estruturas e métodos utilizados na inserção e apresentação das informações adquiridas.

2 REFERENCIAL TEÓRICO

Para o referencial teórico serão abordados conceitos sobre Mineração de Dados e o modelo de inferência CRISP-DM. Será apresentada uma breve introdução a estes de maneira que sejam apresentados os processos, também serão abordadas as técnicas aplicadas na Mineração de Dados bem como alguns dos algoritmos comumente utilizados em sua implementação.

2.2 MINERAÇÃO DE DADOS

A Mineração de dados é um processo crucial para o descobrimento de informações, pois a partir dos dados obtidos tem-se uma vasta gama de informações latentes neles contidas. Da Silva (2016, p. 7) a descreve da seguinte maneira "Mineração de dados é definida em termos de esforço para descoberta de padrões em bases de dados". Observando este conceito, percebe-se que pela descoberta destes padrões é possível a obtenção informações que proporcionem auxílio em tomadas de decisões de negócios ou sua utilização para análise científica, como é proposto a este trabalho.

Nos últimos tempos a mineração de dados vem sendo grandemente utilizada no âmbito empresarial e científico com a finalidade de buscar informações sobre dados específicos que possam auxiliar em aquisição de informações novas e coerentes para a empresa e/ou projeto. Cortês (2002) traz a aplicação de Data Mining no ramo científico e empresarial da seguinte maneira:

> Frequentemente, mineração de dados tem sido considerada e classificada como uma mistura de pesquisas em estatística, inteligência artificial e bancos de dados. Até recentemente, não era reconhecido como um campo de interesse para os estatísticos, sendo mesmo considerado, nesta área, como uma área de pesquisa 'pouco relevante'. Devido à sua importância prática, entretanto, o campo tem emergido como uma área de crescimento acentuado e de elevada importância, destacando-se pelo surgimento de diversos congressos científicos e produtos comerciais. (Cortes, 2002)

A aplicação de MD consiste na implementação de técnicas pré-definidas que propõe o retorno de suas análises em formatos distintos. Segundo o conceito apontado por Tan (2009, p.3) A Mineração de dados consiste na busca e descoberta de padrões de forma automática a partir de grandes bases de dados. Desta forma para esta descoberta automática são utilizados

algoritmos computacionais para extração e análise destas informações. Da Silva também descreve a Mineração de dados da seguinte forma:

> Trata-se, portanto, da aplicação de técnicas, implementadas por meio de algoritmos computacionais, capazes de receber, como entrada, um conjunto de fatos ocorridos no mundo real e devolver, como saída, um padrão de comportamento, o qual pode ser expresso, por exemplo, como uma regra de associação, uma função de mapeamento ou modelagem de um perfil. (DA SILVA; PERES; BOSCARIOLI, 2016, p. 7)

As técnicas (algoritmos) de MD implementam, em sua generalidade, conceitos de Inteligência Artificial e Aprendizado de Máquina. "Tradicionalmente, os métodos de mineração de dados são divididos entre aprendizado supervisionado (preditivo) e não-supervisionado (descritivo)" (CAMILO; SILVA, 2009, p. 10). Conforme os objetivos deste trabalho serão abordados os conceitos e aplicação dos algoritmos de Classificação, Associação, Clusterização e Seleção de Atributos nos tópicos posteriores.

2.1.1 Classificação

A tarefa de classificação consiste na definição de classes a partir de informações recorrentes entre os dados analisados. Conforme Bartolomeu (2002) é uma das tarefas mais comuns da Mineração de Dados, consistindo da localização de propriedades comuns entre um conjunto de dados em uma base e classificação desses dados em classes pré-definidas, seguindo o modelo estipulado. Alguns dos exemplos de aplicação da tarefa de classificação são: atribuir palavras chaves a artigos jornalísticos, classificar pedidos de créditos como baixo, médio e alto risco; esclarecer pedidos de seguro fraudulentos, entre outros.

Devido a necessidade do acompanhamento e do "treinamento" do algoritmo, a classificação encaixa-se como sendo uma tarefa de mineração de dados supervisionada. Segundo Romero e Ventura (2007) a classificação está relacionada a predição de forma que a classificação prevê rótulos de classes enquanto a predição foca em funções de valor contínuo.

Olson e Delen (2008) abordam que na classificação os métodos são destinados ao aprendizado de diferentes funções que mapeiam cada item do conjunto de dados selecionados em um dos conjuntos de classes pré-definidos. Ao se passar o conjunto de classes pré-definidas, número de atributos e conjunto de dados para "treinamento" do algoritmo, este é capaz de predizer a classe de um dado não classificado do conjunto de treinamento. A Tabela

1 apresenta uma exemplificação do conjunto de treinamento e o conjunto a ser aplicado o algoritmo para predição.

Tabela 1- Conjuntos de treinamento e predições geradas a partir de uma base de dados clínica.

Idade	**Frequência Cardíaca**	**Pressão sanguínea**	**Problema cardíaco**
65	**78**	**150/70**	**Sim**
37	**83**	**112/76**	**Não**
71	**67**	**108/65**	**Não**

Conjunto de predição

Idade	**Frequência Cardíaca**	**Pressão sanguínea**	**Problema cardíaco**
43	**98**	**147/89**	**?**
65	**58**	**106/63**	**?**
84	**77**	**150/65**	**?**

Fonte: Adaptado de Vozniak e Viana (2007)

A Tabela 1 apresenta dois conjuntos criados a partir de uma base de dados clínica, em que são utilizados os dados do paciente como idade, frequência cardíaca, pressão sanguínea e se este possui problemas cardíacos. Com base no conjunto de treinamento o algoritmo será capaz de predizer a condição de o paciente ter problemas cardíacos como “sim” e “não” pelo algoritmo para o caso deste possuir ou não problemas cardíacos. Diante disso, é possível observar a possibilidade de aplicação da classificação em bases de dados selecionadas a fim de que os dados sejam agrupados de acordo com os rótulos das classes pré-definidas na etapa de aprendizado.

Para a classificação dos dados, tendo como base as classes, existem técnicas matemáticas que são comumente utilizadas para a aplicação do algoritmo. As técnicas de árvores binárias de decisão, redes neurais, programação linear e estatísticas são as mais utilizadas na aplicação do algoritmo (OLSON; DELEN, 2008, p.16). Estas técnicas são

computacionalmente aplicáveis pelo algoritmo para a predição das classes e obtenção dos resultados.

2.1.2 Clusterização

No agrupamento os dados analisados são separados em subgrupos ou clusters. “Seu objetivo é formar grupos baseados no princípio de que esses grupos devem ser o mais homogêneo em si e mais heterogêneo entre si” (CÔRTES; PORCARO; LIFSCHITZ, 2002, p. 8). A diferença entre a aplicação do agrupamento e da classificação está no fato de que o algoritmo de agrupamento não utiliza classes predefinidas para agrupar os dados analisados, sendo agrupados com base em similaridades. “Na segmentação não há classes nem exemplos predefinidos. Os registros são agrupados de acordo com a semelhança, e a partir daí o significado será determinado.” (BARTOLOMEU, 2012)

A análise clusterizada propõe a execução do algoritmo sobre dados não agrupados e utiliza técnicas automatizadas para colocar os dados em grupos (OLSON; DELEN, 2008, p.17). Por este fato e por não requerer conjuntos de treinamentos para o seu funcionamento a Clusterização é considerada uma técnica de mineração de dados não supervisionada. Kantardzic (2011) define a análise clusterizada a clusterização propõe o estudo formal de métodos e algoritmos para agrupamentos naturais de objetos de acordo com métricas, características intrínsecas ou similaridades percebidas nos dados.

Ainda segundo Kantardzic (2011), a clusterização é observada como um conjunto de metodologias para classificação automática de amostras em grupos utilizando de métricas de associação de forma que estas amostras agrupadas em um grupo sejam similares, enquanto amostras pertencentes a grupos distintos são diferentes. Olson e Delen (2008) afirmam que a clusterização compartilha uma área metodologia comum a classificação, em que certa parte dos modelos matemáticos recomendados para a análise classificativa podem ser utilizados para a análise clusterizada.

A partir dos dados de entrada a aplicação do algoritmo de clusterização gera amostras de clusters com base nas similaridades encontradas. A Tabela 2 apresenta uma adaptação da tabela de exemplo proposta por Kantardzic (2011), a qual ilustra um exemplo simples de informação clusterizada para nove consumidores distribuídos em três clusters.

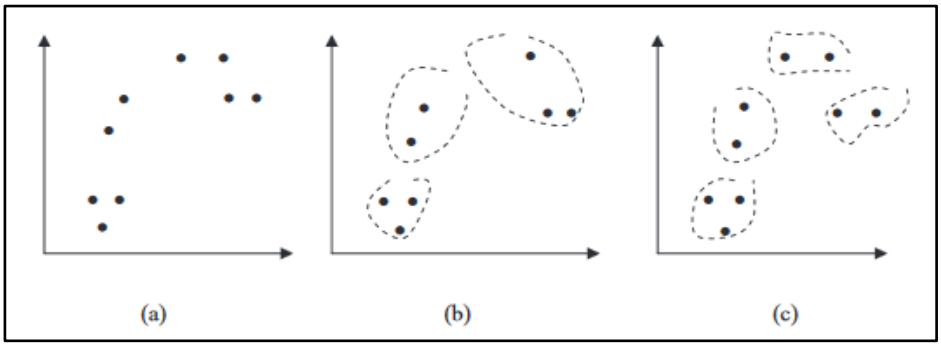
Tabela 2 - Conjunto simples de clusters consistindo de objetos similares.

	Número de Itens	Preço
Cluster 1	2	1700
	3	2000
	4	2300
Cluster 2	10	1800
	11	2100
	12	2500
Cluster 3	2	100
	3	200
	3	350

Fonte: Adaptado de Kantardzic (2011)

A Tabela 2 utiliza de dois dados relacionados aos consumidores, sendo o número de itens comprados e o preço do produto, para poder gerar a divisão por clusters com base na similaridade. Ao observar-se a tabela é possível constatar a forma em que os clusters foram agrupados, sendo que no Cluster 1 são os consumidores que compraram poucos itens com preço alto, já o Cluster 2 emprega os consumidores que fizeram a aquisição de muitos itens com preços altos e por fim o Cluster 3, que compõe os consumidores que adquiriram poucos itens com preço baixo. A Figura 1 ilustra a formação de clusters a partir de pontos de entrada como dados.

Figura 1 - Exemplo de análise clusterizada de pontos em um espaço 2D.



Fonte: Kantardzic (2011)

O gráfico (a) da Figura 1 apresenta um conjunto de pontos de entrada em que se criam diversos clusters compostos por dois e três pontos que partilham de certa similaridade. Os gráficos b) e c) ilustram dois exemplos de agrupamento destes pontos, sendo o exemplo b) com três clusters gerados e o exemplo c) com quatro clusters gerados.

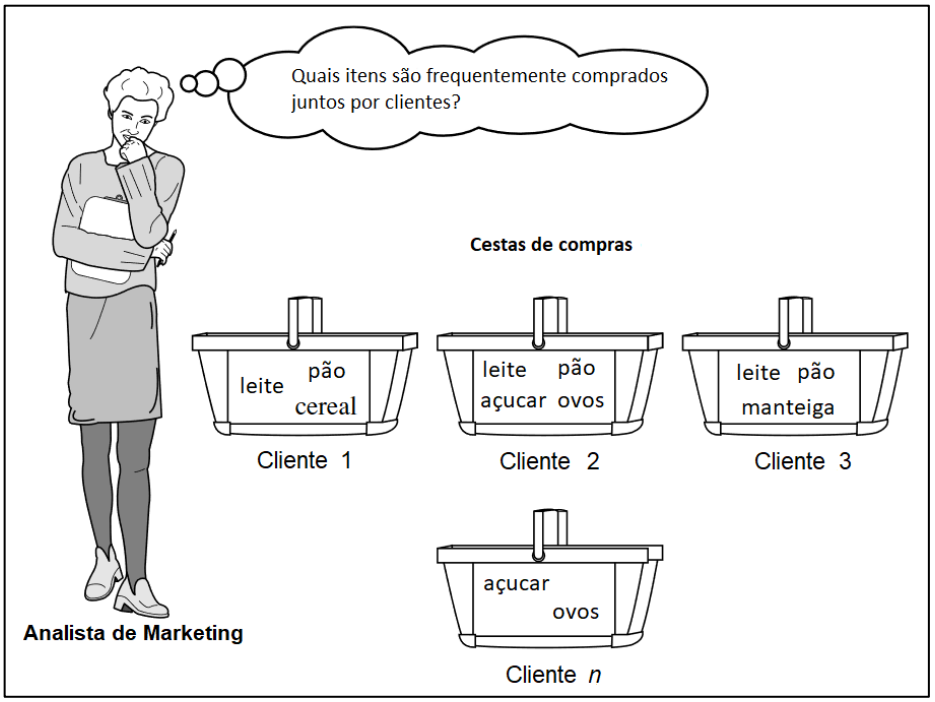
A análise clusterizada pode ser aplicada em diversas áreas de estudo com a finalidade de agrupar dados com base em similaridades destes, como área empresarial, saúde, engenharias e etc. Bijuraj (2013, p.3) exemplifica seu uso na análise de vendas de produtos de uma companhia, a fim de se identificar qual está sendo mais vendido e qual está sendo menos vendido com base nos agrupamentos resultantes da aplicação do método.

2.1.3 Associação

De acordo com Kantardzic (2011, p.281) a associação é uma das maiores técnicas de data mining e é comumente utilizada para a descoberta de padrões de forma não supervisionada. Esta metodologia busca em grandes bases de dados aspectos que ajudem a compreender os padrões. Os algoritmos executados buscam relações entre os dados, de modo que são verificados os eventos que ocorrem de forma concorrente a fim de se alcançar melhores resultados (PELEGRIN et. al, 2005, p.2).

O principal resultado da utilização dos algoritmos são as regras de associação, que consistem na ocorrência de determinado relacionamento entre itens de um conjunto de dados. De acordo com Han, Kamber e Pei (2012) um exemplo típico da análise associativa é a análise da cesta de compras (*market basket analysis*) representada na Figura 2, que propõe analisar os hábitos de compra dos consumidores de um mercado através de associações encontradas entre os diferentes itens que os clientes colocam em suas cestas de compras.

Figura 2 - Análise da cesta de compras.



Fonte: Han, Kamber e Pei (2012)

As descobertas resultantes desta análise podem auxiliar no desenvolvimento de estratégias de vendas ao se avaliar quais itens são comprados em conjunto pelos consumidores. Como exemplo da aplicação desta técnica pode-se citar a verificação de quais itens adicionais e com que frequência são comprados junto com o produto leite. O resultado dessa análise possibilita aos estabelecimentos utilizarem de estratégias de posicionamentos dos produtos para aumentar seus ganhos (HAN; KAMBER; PEI, 2012).

Ao buscar associações existentes entre os dados, deve-se calcular a incidência de um conjunto de itens na cesta e comparar este valor com um limite pré-determinado chamado suporte mínimo. Nisso, os conjuntos de itens que possuírem uma quantidade de incidência maior que o valor superior ao suporte mínimo são classificados como grandes conjuntos de itens que serão utilizados para a construir as regras de associação (SOUZA FILHO, 2004).

Com base nos conjuntos de itens estabelecidos, a associação poderá inferir relações entre estes itens e gerar regras de associação que as represente. “A tarefa de associação consiste em identificar quais atributos estão relacionados. Apresentam a forma: SE atributo X ENTÃO atributo Y” (CAMILO; SILVA, 2009, p. 10). Seguindo esta forma Liu (2007) propõe a formalização das regras de associação da seguinte maneira. Sejam $I = \{i_1, i_2, \ldots i_m\}$ um conjunto de itens e T um conjunto de transações, ou seja, uma base de dados, onde cada transação t_i corresponde a um conjunto de itens em que $t_i \subset I$. Com isso, uma regra de

associação implica na fórmula: $X \Rightarrow Y$, *onde* $X \subset I$, $Y \subset I$, *e* $X \cap Y = \emptyset$, em que *X* ou *Y* são conjuntos de itens chamados de *itemsets*.

Uma regra de associação possui suporte e confiança, que são métricas que auxiliam a medir o nível de interesse da regra gerada. Estas métricas refletem a usabilidade e a certeza de regras geradas (HAN; KAMBER; PEI, 2012). Segundo Liu (2007) o suporte de uma regra $X \Rightarrow Y$ consiste na quantidade de incidências que uma regra foi aplicada sobre um conjunto de dados, ou seja, a porcentagem de transações em T que contenham $X \cup Y$. Sendo assim computado pela fórmula apresentada na Figura 3.

Figura 3 - Fórmula de suporte para regra de associação

$$\text{suporte} = \frac{(X \cup Y).\text{contador}}{n}$$

Fonte: Liu (2007)

Conforme a Figura 3, o suporte é calculado pela quantidade de incidência (contador) de $X \Rightarrow Y$ sobre *n*, em que *n* consiste na quantidade de transações. Já a confiança de uma regra baseia-se no quão forte esta regra é, analisando a incidência de um item específico sobre outro. Em outras palavras, a confiança de uma regra A⇒ B representa, dentre as transações que contenham o item A o percentual de transações que também contenham o item B (GONÇALVES, 2004) e é computada pela fórmula apresentada na Figura 4.

Figura 4 - Fórmula de confiança para regra de associação

$$\text{confiança} = \frac{(X \cup Y).\text{contador}}{X.\text{contador}}$$

Fonte: Liu (2007)

Para o cálculo da confiança seguindo a fórmula proposta na Figura 4 leva-se em conta a incidência de X ∪ Y (contador) sobre a quantidade de transações que possuem o item X. De acordo com Gonçalves (2004), um dos modelos mais comuns para a mineração de dados por associação é o modelo Suporte/Confiança. Este modelo propõe encontrar as regras de associação que possuam um suporte e confiança maiores ou iguais a um suporte mínimo e confiança mínima estipulados pelo analisador. Com isso são apresentadas apenas as regras

que obedecem ao exigido pelo usuário, limitando assim o número de regras resultantes da aplicação de um algoritmo.

Um dos algoritmos mais utilizados para esta tarefa é o Apriori, introduzido por Agrawal et al. (1994) e utilizado para gerar regras de associação com base em conjuntos de itens de uma grande base de dados. "O algoritmo emprega busca em profundidade e gera conjuntos de itens candidatos (padrões) de *k* elementos a partir de conjuntos de itens de *k - 1* elementos." (VASCONCELOS; CARVALHO, 2004, p. 11). Nele os padrões que geram repetições são eliminados a fim de se evitar regras de associação recorrentes, e a partir dos itens candidatos são recuperados os itens frequentes. O Apriori trabalha em torno de dois passos que são abordados por Liu (2007, p. 20) como:

1. **Gerar todos os conjuntos de itens frequentes:** Um conjunto de itens frequente é o conjunto de itens cujo suporte ficou acima do MinSup.
2. **Gerar todas as regras de association confiáveis a partir dos conjuntos de itens frequentes:** Uma regra de associação confiável é a regra em que a confiança está acima do MinConf.

A partir destes passos o algoritmo consegue gerar e validar as regras de associação que correspondam com o requerido pelo usuário para análise. A Figura 5 exemplifica a codificação do algoritmo de Apriori por meio de um pseudocódigo.

Figura 5 - Pseudocódigo do algoritmo Apriori

```
   F1 ←{Conjuntos de itens freqüentes de tamanho 1} /* Na
      primeira passagem k = 1                          */
1  para k = 2; Fk−1 != vazio; k++ faça
   |  /* Na segunda passagem k = 2                     */
2  |  Ck ← apriori-gen (Fk−1) /* Novos candidatos      */
3  |  para todo transação t ∈ T faça
4  |  |  Ct ← subconjunto (Ck, t) /* Candidatos contidos
   |  |     em t                                       */
5  |  |  para todo candidato c ∈ Ct faça
6  |  |  |  c.contagem++
7  |  |  fim
8  |  |  Fk ← {c ∈ Ck|c.contagem ≥ MinSup}
9  |  fim
10 fim
11 Resposta F ← Reunião de todos os Fk
```

Fonte: Carvalho (2004, p. 11)

A Figura 5 apresenta de forma comentada um pseudocódigo correspondente ao algoritmo Apriori. Abaixo é apresentada uma legenda das variáveis contidas no pseudocódigo.

- F_k - Consiste no conjunto de itens frequentes de tamanho k que atende ao MinSup. Cada componente deste conjunto dispõe de dois campos, sendo o primeiro a contagem de itens contidos e o segundo trata-se do contador para o suporte.
- C_k - Trata-se do conjunto de itens candidatos de tamanho k. Assim como F_k cada componente do conjunto dispõe de dois campos sendo um para o conjunto de itens e outro do contador para o suporte, respectivamente.

Apesar de ser um algoritmo bastante difundido na análise de dados, sua aplicação não é algo simples de ser implementada. "A implementação do algoritmo Apriori envolve estruturas de dados e técnicas de programação sofisticadas [...]" (LIU, 2007, p.24). Contudo, existem diversas plataformas de mineração de dados que já implementam o Apriori e disponibilizam de forma fácil ao usuário, sendo necessário apenas informar os conjuntos de dados a serem geradas as associações, o MinSup e o MinConf. Como exemplo a isso tem-se o RStudio que tem disponível o pacote *arules*[1] para análises associativas e que é baseado no algoritmo Apriori.

2.2 DATA WAREHOUSE

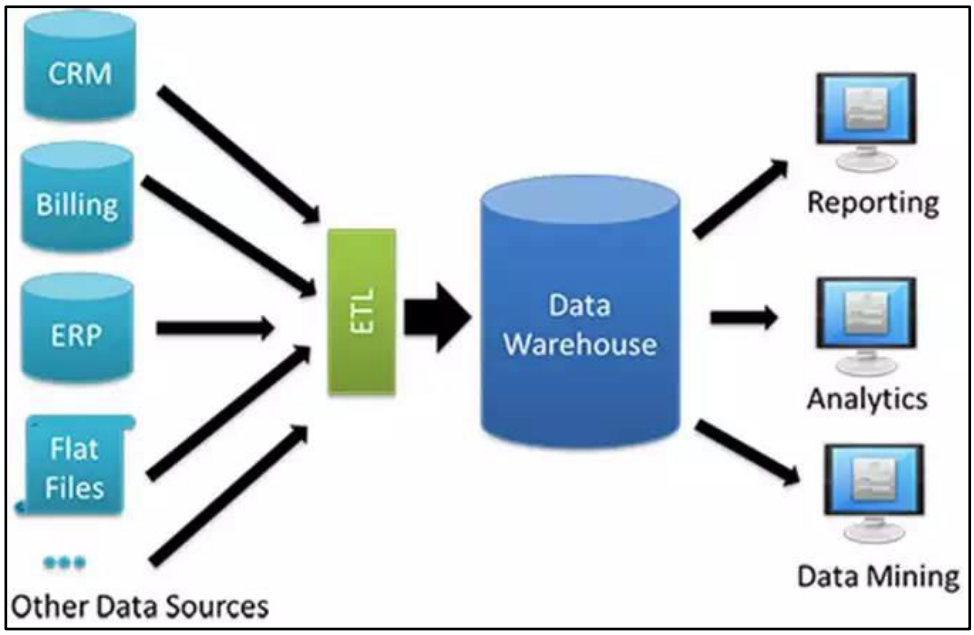
"Um *warehouse* (armazém) é uma coleção de dados, orientado a um assunto, integrado, tempo-variante e não volátil, para suporte ao gerenciamento dos processos de tomada de decisão" (INMON, 2002, p.31). O principal objetivo de um *Data Warehouse* (DW) consiste no fornecimento de dados históricos armazenados, de forma que seja possível coletar informações necessárias para determinadas aplicações.

Os *Data Warehouses* surgiram de um conceito acadêmico, que com o amadurecimento dos sistemas empresariais, fez com que as necessidades de análise de dados crescessem paralelamente, sendo assim necessário que houvessem análises dos dados de forma a obter informações válidas e concretas para a tomada de decisões que afetariam as empresas e seu estabelecimento no mercado. Vale ressaltar que assim como o descrito por Inmon (2002), os dados armazenados por um Data Warehouse são utilizados apenas para leitura, sendo assim não é possível alterá-los, tornando-os uma fonte de informações confiáveis.

[1] Documentação: https://cran.r-project.org/web/packages/arules/arules.pdf

De acordo com Cavalcanti, Fell e Dornelas (2005), no DW existem somente duas operações, a carga inicial e as consultas dos *front-ends* aos dados. Após serem integrados e transformados, os dados são carregados em bloco para o data warehouse, para que estejam disponíveis aos usuários para acesso. Deste modo, com a base de dados já estruturada e preenchida com os dados, cabe aos desenvolvedores *front-end* a utilização destes dados para apresentação. Segundo Cielo (2008) a partir do cruzamento de informações fornecidas pelo DW é possível uma análise e tomada de decisões utilizando dados concretos e não intuições. A Figura 6 apresenta uma representação da arquitetura de um Data warehouse.

Figura 6 - Processo para a montagem de um data warehouse.



Fonte: Monitis[1]. (2016)

O primeiro passo a ser tomado quanto a criação do DW é a modelagem do mesmo. Diante disto é preciso a criação de um modelo conceitual que abranja as tabelas, as quais os dados serão armazenados. Para a confecção dos dados é preciso fazer o uso de uma ferramenta ETL que fará a extração destes dados de sua base de origem, fará o tratamento necessário, como padronização das informações, e por fim irá fazer a carga dos dados obtidos no banco de dados ou *data warehouse*. Para o acesso aos dados/informações dispostas no *data warehouse* é em geral utilizado técnicas OLAP, que também são comumente utilizadas para análises destes dados.

2.2.1 Modelagem do DW

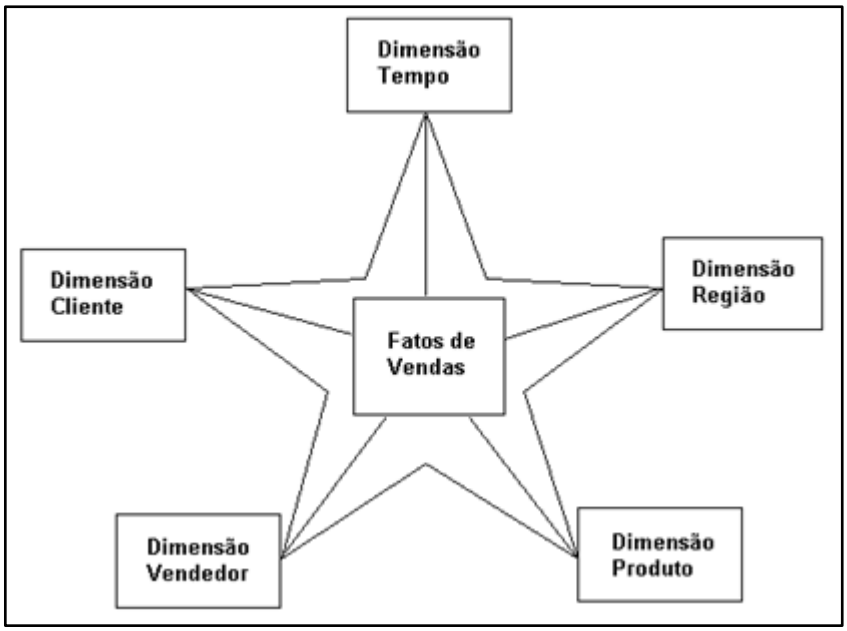
"O modelo dimensional para construção de banco de dados para Data Warehouse é uma forma de modelagem onde as informações se relacionam de forma que pode ser

representada como um cubo." (MOREIRA, 2006). O modelo dimensional proporciona a visualização de dados abstratos de forma simples e realizar o cruzamento entre informações diversas de forma muito eficaz. A criação do modelo dimensional é feita fundamentalmente pela utilização das tabelas fato e dimensão aplicadas dentro de um tipo específico de modelo dimensional.

- Tabela Fato: Trata-se da tabela principal de um modelo dimensional, nela são armazenadas as medições do negócio de interesse empresarial tais como quantidade de determinado produto vendido e valor da venda. Além destas informações a tabela fato contém as chaves para as tabelas dimensões.
- Tabela Dimensão: "As tabelas dimensões compõem as descrições textuais sobre cada um dos elementos que fazem parte do processo." (MOREIRA, 2006). Nelas estão contidos os atributos que descrevem as características que possam ser úteis para análises feitas a partir do *data warehouse*.

No desenvolvimento do DW existem diversos modelos de estruturação, o mais comumente utilizado é o modelo estrela por ser de fácil compreensão e oferecer uma melhor performance durante a execução de consultas. A Figura 7 apresenta o modelo Estrela.

Figura 7 - Modelo Estrela



Fonte: Machado² (2000)

Na Figura 7 é possível observar a representação de um modelo estrela em um *data warehouse* em um contexto comercial, onde as tabelas dimensão (pontas) são características

sobre as vendas e a tabela fato (centro) armazena as informações principais sobre as vendas. O que favorece o uso do "modelo estrela" em comparação ao "floco de neve" é que no modelo estrela não há normalização de dados, facilitando o cruzamento de informações.

2.3 MODELO DE REFERÊNCIA CRISP-DM

O CRISP-DM é uma metodologia para Mineração de Dados que propõe aumentar a taxa de sucesso de processos de Data Mining. Segundo a IBM (2017) "CRISP-DM, que significa Processo Padrão de Vários Segmentos de Mercados para Mineração de Dados, é uma forma comprovada pelo mercado para orientar seus esforços de mineração de dados". O CRISP-DM propõe padronizar as fases e atividades relacionadas a Mineração de Dados, provendo uma visão geral sobre o ciclo de vida de um projeto envolvendo data mining (CHAPMAN et al., 2005, p.10).

O modelo proposto no CRISP-DM consiste de seis fases que são organizadas de maneira cíclica. Segundo Moro (2011, p.2) "CRISP-DM define um projeto como um processo cíclico, em que várias iterações podem ser usadas para permitir um resultado ajustado aos objetivos de negócio". Além disso, ao se trabalhar como o modelo, é possível a transição entre fases sem seguir um fluxo, sendo assim um modelo unidirecional.

Segundo Chapman, et al. (2005, p.6) "a metodologia CRISP-DM é descrita em termos de um modelo de processo hierárquico, consistindo de uma série de tarefas descritas em quatro níveis de abstração (do geral ao específico): fases, tarefas genéricas, tarefas especializadas e processos de instância". Estes níveis operam com base nas fases propostas do CRISP-DM de modo que estas estão incluídas em cada um destes níveis. Os quatro níveis de abstração serão melhor descritos nos tópicos a seguir:

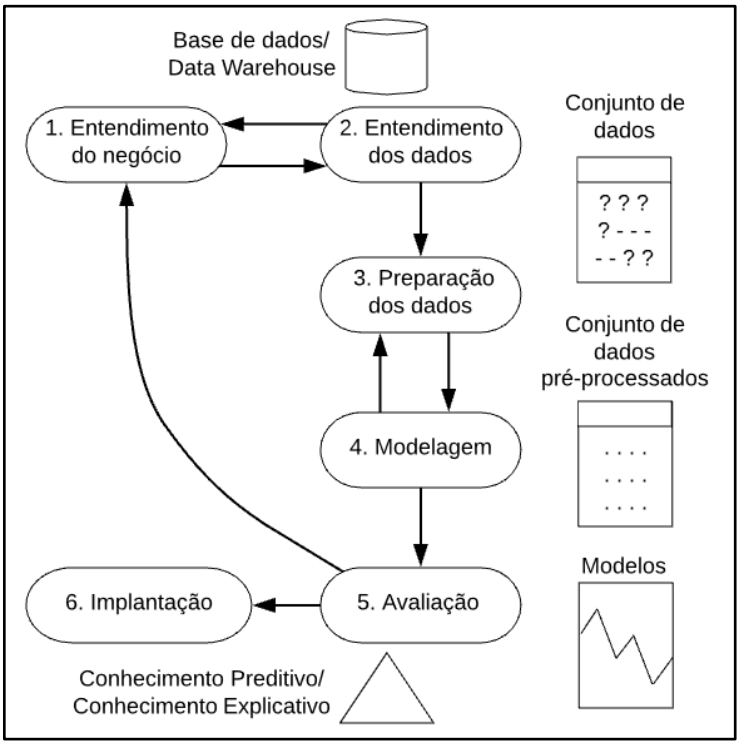
- Fases: é o nível com teor mais geral, consiste em estruturar o processo de Data Mining em diversas tarefas em que cada fase contém um conjunto de tarefas do nível genérico. Deste modo encontra-se a necessidade do entendimento do negócio e dos dados para delimitar os objetivos do projeto e como o processo será estruturado a partir das fases seguintes.
- Tarefas Genéricas: É o segundo nível dentre os quatro, é considerado genérico por se dedicar a tentar atingir as possíveis situações que podem decorrer da mineração de dados. Nele são especificadas ações que sejam o mais estáveis, completas e que atendam aos requisitos do projeto. Neste nível é feito a preparação dos dados, a decisão e aplicação dos algoritmos que virão a ser utilizados a fim de alcançar os

objetivos estabelecidos para o projeto. Como produto da aplicação dos algoritmos são gerados modelos de resultados para serem analisados e avaliados.

- Tarefas Especializadas: É o nível responsável pela descrição das ações aplicadas nas tarefas propostas anteriormente e avaliação dos resultados oriundos das decisões tomadas e ações executadas. Esta etapa aborda a fase de avaliação do CRISP-DM de modo que os resultados da aplicação dos algoritmos de MD (modelos) são avaliados quanto a sua veracidade e integridade. Assim, com esta análise é possível verificar se estes resultados são coerentes e vão de encontro aos objetivos estabelecidos para o projeto.
- Processos de Instância: Neste nível é feito o registro das ações, tarefas, decisões e resultados obtidos dos níveis anteriores. Este nível se enquadra na fase de Implantação do CRISP-DM, de modo que o registro é organizado conforme as tarefas definidas nos níveis superiores e representa de forma clara o que ocorreu no desenvolver do projeto.

Diante disso a Figura 8 apresenta, em formato gráfico, as fases definidas no modelo de referências proposto no CRISP-DM.

Figura 8 - Modelo de processos do CRISP-DM



Fonte: Moro (2011)

Ao se observar o desenho da metodologia proposto por Moro (2011) na Figura 8 é possível observar a estruturação da metodologia e suas fases, de modo que seria a forma

desejada de sua aplicação. As fases propostas na metodologia do CRISP-DM serão abordadas nas seções dispostas a seguir.

- **Entendimento do negócio**

O entendimento do negócio foca na compreensão do objetivo a ser atingido pela Mineração de Dados, sendo de fundamental importância para o desenvolvimento das demais etapas do modelo. “A partir do entendimento do projeto e dos requisitos em uma perspectiva de negócio, converte-se esse conhecimento em uma definição de problema de data mining e em um plano preliminar para alcançar os objetivos” (CHAPMAN et al., 2005, p. 10).

- **Entendimento dos dados**

O entendimento dos dados consiste em compreender os dados que estão sendo analisados e assim identificar o conjunto de dados relevante à proposta. De acordo com Camilo e Silva (2009) “As fontes fornecedoras dos dados podem vir de diversos locais e possuírem diversos formatos”. Com base nos dados adquiridos cabe ao analista separá-los e verificar se estes se encaixam no modelo proposto de trabalho no entendimento do negócio. Segundo Olson e Delen (2008) esta fase do processo pode incluir coleta inicial de dados, descrição dos dados, exploração dos dados (no âmbito de observar sua estruturação) e a verificação de qualidade dos dados. Além disso, os autores também abordam a possibilidade de utilização de clusterização com a intenção de identificar padrões nos dados de modo que já possam ser observados possíveis algoritmos a serem aplicados e delimitar o foco em quais dados poderão fornecer resultados que atendam aos objetivos.

- **Preparação dos dados**

A preparação dos dados consiste na formatação e transformação dos dados de modo a padronizá-los. Segundo Olson e Delen (2008) o propósito da preparação dos dados é limpar os dados selecionados de modo a obter-se melhor qualidade, tendo em vista que alguns dos dados selecionados podem seguir diferentes padrões por conta de serem coletados de diferentes fontes. Para isso, é comum a utilização de ferramentas computacionais para auxiliar na execução da limpeza dos dados de forma correta e eficiente. De acordo com Hiragi (2009) é nesta fase em que ocorre a seleção de atributos, tratamento de valores faltosos, erros em partes dos dados, formatação e padronização e a divisão dos dados em, ao menos, um conjunto de treinamento e um conjunto de teste. Desta forma, os resultados desta etapa serão os conjuntos de dados prontos para serem utilizados na mineração de dados.

- **Modelagem**

Na modelagem são selecionados e aplicados os algoritmos de mineração de dados de modo a alcançar os resultados esperados. “Nesta fase, algoritmos de aprendizagem de máquina mais adequados a cada cenário são configurados para construir modelos aderentes e compatíveis com os dados preparados” (CARVALHO, 2016, p. 11). Conforme Chapman et al. (2005) comumente existem diversas técnicas que podem ser aplicadas para um mesmo tipo de problema de MD. Nisso, algumas técnicas necessitam de requisitos específicos nos dados formatados, o que torna necessário retornar a etapa de preparação dos dados. Deste modo, com a aplicação das tarefas de MD sobre os dados, geram-se os modelos que serão avaliados no contexto do projeto.

- **Avaliação**

“Considerada uma fase crítica do processo de mineração, a avaliação é a etapa que conta com a participação de especialistas nos dados, conhecedores do negócio e tomadores de decisão. ” (CAMILO; SILVA, 2009, p. 5). Como forma de auxílio ao processo da avaliação são utilizados gráficos para analisar e visualizar os resultados obtidos em cada uma das etapas anteriores, deste modo, para garantir a confiabilidade dos modelos é indicada a realização de testes e validações dos modelos construídos. É nesta etapa que os conhecimentos dos dados minerados são assimilados e para isso há duas questões essenciais. A primeira se encontra em como reconhecer o valor dos novos padrões descobertos para o negócio. A segunda é qual ferramenta de visualização deve ser utilizada para apresentar os resultados da mineração de dados (Olson e Delen, 2008). Deste modo é relevante ressaltar que uma boa interpretação e avaliação dos modelos acarreta em informações produtivas para o negócio e em contrapartida uma interpretação pobre e avaliação ruim levam à perda de informação útil.

- **Implantação**

Na implantação os resultados do projeto de MD são colocados em uso e apresentados aos envolvidos. “O resultado do estudo dos resultados da mineração de dados deve ser reportado aos patrocinadores, visto que este possui novos conhecimentos descobertos, que necessitam de estar bem atados aos objetivos originais do projeto de data mining. ” (OLSON e DELEN, 2008, p. 10). Segundo Hiragi (2009) “A colocação em uso pode ser vista como utilizar resultados obtidos pela aplicação (a um novo conjunto de dados) do modelo selecionado para apoiar uma tomada de decisão por parte do decisor que o utiliza”. Diante disso, a implantação dos resultados do projeto pode atuar como fator crucial sobre os negócios dos envolvidos ao auxiliar na gerência do negócio. Desta forma Chapman et al (2005) ressalta que dependendo dos requisitos, a fase de implantação pode ser tão simples como gerar um

relatório ou tão complexa quanto implementar um processo repetitivo de mineração de dados por toda a empresa e que em muitos casos é o cliente, e não o analista de dados, que dá realiza os passos da implantação.

3 METODOLOGIA

Esta seção apresenta a metodologia aplicada a este trabalho, abrangendo os materiais e procedimentos que foram utilizados no desenvolvimento desta proposta.

3.1 OBJETO DE ESTUDO

O presente trabalho tem por propósito o desenvolvimento de uma plataforma de visualização dos resultados obtidos da aplicação de técnicas e algoritmos de Data Mining sobre os dados de respostas dos estudantes de graduação em provas do ENADE. Este processo foi baseado nas fases do modelo CRISP-DM, de modo que fosse possível avaliar o desempenho dos acadêmicos em campos de estudo. Os dados a serem utilizados para esta pesquisa foram extraídos dos microdados do ENADE dispostos no website do Inep em arquivos no formato .csv, que contém informações sobre os acadêmicos, suas respostas e os gabaritos respectivos para as provas de cada área de todo o país.

3.2 MATERIAIS

Foi realizado o levantamento do material bibliográfico relevante para a constituição do referencial teórico, incluindo livros, artigos, publicações científicas e demais conteúdos digitais. A partir disso foi construído o referencial teórico sobre Data Mining contendo suas técnicas e métodos, bem como, o CRISP-DM e suas fases. Estes estão conceituados na seção de Referencial Teórico, tendo como principal foco as partes necessárias para o entendimento do trabalho.

3.2.1 Tecnologias/Ferramentas

Para que a proposta deste trabalho fosse atendida, foram utilizados o Microsoft Excel em conjunto à linguagem de programação Python como ferramentas para tratamento de dados, tendo em vista que estes permitem a limpeza e tratamento dos dados de forma dinâmica e simples. Para a análise foram utilizadas as ferramentas R Studio e Weka, juntamente a linguagem de programação R, onde utilizou-se dos dados em um arquivo no formato .csv que é possível de ser utilizado em ambas as ferramentas. Para o desenvolvimento da api foi utilizado o Django Rest Framework que foi escolhido devido a rapidez e versatilidade no desenvolvimento e consultas à API, pois para alcançar os objetivos deste trabalho o Django Rest foi a ferramenta que mais se encaixou no contexto aplicado. Deste modo, também foi o utilizado o BD nativo do Django, o SQLite que também é de fácil configuração e manipulação através do Django Admin. Para a carga dos dados na base de

dados foi utilizada a ferramenta de ETL Pentaho Data Integration, da empresa Pentaho, que permite o tratamento dos dados de formas customizáveis que se adequassem à proposta deste trabalho, deste modo permitindo o tratamento dos dados de maneira simples e eficiente para se encaixarem nos padrões do BD.

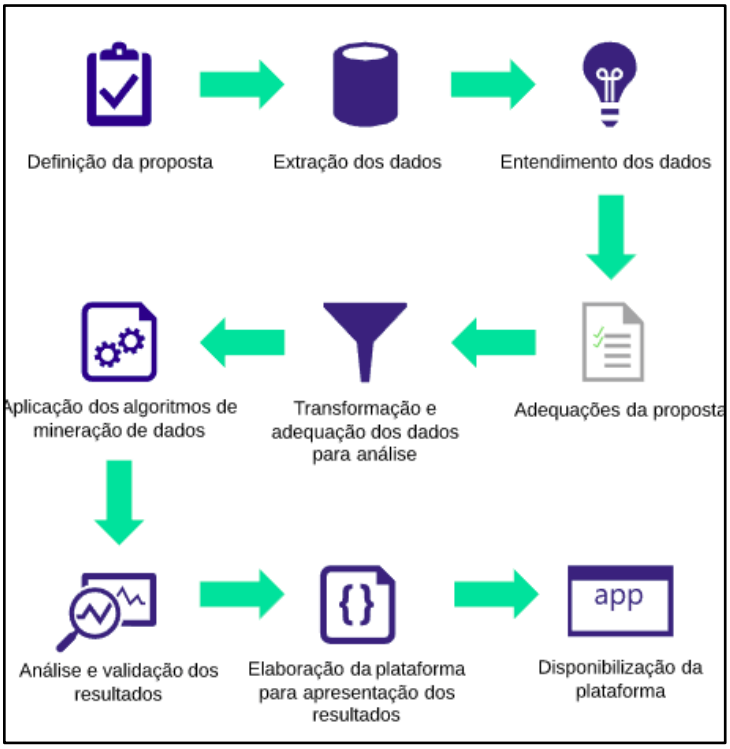
De modo a apresentar os resultados, foram utilizadas técnicas de programação Front-End, a fim de que seja criada uma página Web para exibição dos resultados. Para a codificação da página e das demais tarefas será utilizado o editor de código Visual Studio Code. A fim de criar e exibir os resultados contidos na base de dados em informação visual, serão utilizados os frameworks Angular 5 e Bootstrap 4, junto a seus plugins e bibliotecas que proporcionam a criação de gráficos mediante a dados fornecidos.

3.3 PROCEDIMENTOS

Durante o projeto foram consideradas as respostas ao questionário específico dos estudantes de Bacharelado dos cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação das Instituições de Ensino Superior brasileiras. Esta parcela de discentes foi escolhida pelo fato deste conjunto de discentes estar presente em todas as aplicações das provas dentre os anos mencionados e por se tratar do grupo completo de cursos da área de Tecnologia da Informação que faz o ENADE.

As adequações feitas ao modelo para o desenvolvimento deste trabalho são representadas pela Figura 9 e serão abordadas de forma mais detalhada nos tópicos a seguir.

Figura 9 - Fluxograma do desenvolvimento da proposta



- **Definição da proposta:** a primeira etapa deste trabalho consiste no estabelecimento da proposta do trabalho junto ao orientador do projeto. A proposta foi elaborada englobando quais dados foram utilizados, quais algoritmos fossem interessantes de se aplicar para a análise com base nos objetivos do trabalho e estruturar a forma que estes dados podem ser apresentados, de modo que fosse possível apresentá-los de forma dinâmica a fim de proporcionar uma melhor visualização dos resultados obtidos da análise. Para a análise do desempenho dos alunos, pretende-se classificar as perguntas da prova do Enade dentro das principais áreas da Computação. Para isso, a área foi dividida inicialmente de acordo com as grandes áreas existentes no modelo de mapa curricular[2] dos cursos de Computação (CC e SI) do CEULP/ULBRA. Nesse modelo, a partir de 6 grandes áreas é possível encaixar as disciplinas dos cursos e na sequência identificar as áreas de enquadramentos de cada questão da prova do ENADE. A Tabela 3 apresenta essa classificação, que pode passar por adequações ao longo da execução do trabalho.

Tabela 3 - Áreas da Computação

Grandes áreas	Subáreas	Áreas para enquadramento

[2] http://ulbra-to.br/Cursos/Sistemas-de-Informacao/sisinfomap/
http://ulbra-to.br/cursos/Ciencia-da-Computacao/CompMap/

		das questões
Análise e Desenvolvimento	• Engenharia de Software I • Banco de Dados II • Engenharia de Software II • Modelagem de Sistemas de Informação • Linguagem de Programação para a Web • Gerência de Projetos • Desenvolvimento de Sistemas de Informação • Arquitetura de Software • Qualidade e Auditoria de Software	1. Engenharia de Software 2. Banco de Dados 3. Gerência de Projetos 4. Qualidade de Software 5. Arquitetura de Software
Linguagens de Programação	• Algoritmos e Programação I • Algoritmos e Programação II • Linguagem de Programação Orientada a Objetos I • Linguagem de Programação Comercial I • Linguagem de Programação para a Web	
Fundamentos das Ciências exatas	• Fundamentos de Matemática • Cálculo I • Estatística Aplicada • Lógica de Predicados • Matemática Discreta • Geometria Analítica e Álgebra Linear • Cálculo Numérico Computacional	3. Lógica
Fundamentos da Computação	• Algoritmos e Programação I • Introdução à Computação • Arquitetura e Organização de Computadores I • Algoritmos e Programação II • Banco de Dados I • Estruturas de Dados I • Estruturas de Dados II • Sistemas Operacionais I	4. Arquitetura de Computadores 5. Estrutura de Dados 6. Sistemas Operacionais
Teorias da	• Fundamentos de Sistemas de	

Computação	Informação • Paradigmas de Linguagens de Programação • Linguagens Formais • Compiladores	
Tecnologias da Computação	• Sistemas de Informação I • Sistemas de Informação II • Interface Homem-Computador • Redes de Computadores I • Inteligência Artificial I • Computação Gráfica • Sistemas Distribuídos • Computação Paralela • Segurança de Sistemas • Redes de Computadores II	8. Sistemas de Informação 9. Redes de Computadores 10. Inteligência Artificial 11. Sistemas Operacionais 12. Segurança de Sistemas 13. Sistemas Distribuídos

- **Extração dos dados:** com a definição da proposta delimitada, foram levantados a fonte dos dados e os possíveis os dados a serem analisados. Os dados a serem utilizados no desenvolvimento deste trabalho são dispostos para livre acesso pelo Inep[3] em formato de microdados em planilhas .csv para cada ano de aplicação da prova do ENADE. Após a definição da proposta, foram extraídas as respostas dadas nas provas e as informações dos discentes da área de Computação dos dados do ENADE para os anos pré-definidos. Junto a isso, também foram extraídos os gabaritos das provas, de modo que fosse possível a análise sobre a quantidade de acertos nos campos de estudo da Computação.
- **Entendimento dos dados:** para o entendimento dos dados extraídos, estes foram observados de forma a entender sua estrutura e seus relacionamentos. Para isso, foi necessário utilizar dos dicionários de variáveis que são disponibilizados junto às planilhas de cada ano, identificando as propriedades dos atributos contidos nas planilhas de respostas. Também foi identificado com base em qual(is) campo(s) de estudo cada questão foi construída, de modo que, na etapa de análise dos resultados da mineração de dados fosse possível validar os acertos dos estudantes da área de Computação para cada campo de estudo da área.

[3] http://inep.gov.br/web/guest/microdados

- **Adequações da proposta:** as adequações à proposta foram feitas de acordo com a fase de entendimento dos dados, de maneira que quaisquer informações que possam induzir a alterações na proposta fossem discutidas e ponderadas para verificação e validação da aplicação no contexto do projeto.
- **Transformação e adequação dos dados para análise:** logo após as adequações serem ponderadas, os dados foram transformados e adequados de modo que tornasse possível a aplicação dos algoritmos de mineração de dados. Para isso foram verificados os formatos atuais de visualização das respostas às questões e o gabarito para cada ano apresentados nas planilhas, com essa verificação estes foram padronizados a apenas um formato de dados, de maneira a evitar conflitos na aplicação dos algoritmos e alterações nos resultados da análise dos dados. Dentro deste contexto foram utilizadas a ferramenta Microsoft Excel e a linguagem de programação Python para otimizar o processo, de modo que as transformações e adequações necessárias fossem feitas de forma rápida e dinâmica. Deste modo, os dados foram padronizados de maneira a remover inconsistências nas respostas dos estudantes e no gabarito da prova.
- **Aplicação dos algoritmos de mineração de dados:** com a transformação e padronização dos dados estes encontram-se prontos para a mineração. Diante disso, foram aplicados os algoritmos abordados anteriormente, para obtenção de resultados contendo novos padrões de análise. Para o algoritmo de clusterização foram observados clusters resultantes com base nos campos de estudos da Computação, de forma que as respostas pudessem ser agrupadas de acordo com estes campos e assim fosse possível executar as análises dos resultados com base nos padrões recém descobertos. A aplicação dos algoritmos sobre os dados foi feita utilizando da ferramenta RStudio, em conjunto a linguagem de programação R, e a ferramenta Weka de maneira que fossem obtidos resultados que permitam a análise e posterior validação do modelo. Estas ferramentas executam o algoritmo com base num conjunto de dados de entrada e podem apresentar o resultado de diversas formas, como gráficos; regras de produção; textos e etc; ficando assim a critério do utilizador escolher qual forma será empregada. Para facilitar o processo de análise, a apresentação dos resultados foi configurada de forma gráfica.
- **Análise e validação dos resultados:** os resultados obtidos da utilização dos algoritmos foram analisados, de forma que estes fossem ponderados e validados com

base nos objetivos da proposta. Deste modo estes foram considerados para observar o desempenho dos discentes dos cursos de Computação de acordo com seu curso e campo de estudo com maior acerto. Com base nessa análise foi possível mensurar as taxas de acerto e taxas de erros de questões dos alunos de graduação, o que possibilitou fazer-se um levantamento de quais são os campos de estudo com maior taxa acerto por curso da área de Computação. Esta análise foi efetuada com o acompanhamento do orientador e do especialista do domínio na instituição CEULP/ULBRA, que auxiliou na validação das informações coletadas e no processo analítico destes resultados. Desta forma, com a análise foram obtidas novas informações válidas e coerentes sobre os dados.

- **Elaboração da plataforma para apresentação dos resultados:** utilizando-se de técnicas computacionais e frameworks de desenvolvimento web, foi implementada uma plataforma em que são apresentados os resultados obtidos pela aplicação dos algoritmos e a análise dos resultados de forma visual. A plataforma utiliza gráficos como principal forma de apresentação do conteúdo, de modo que seja agradável, intuitiva e de fácil entendimento pelo público. Estes gráficos apresentaram as informações e resultados obtidos da aplicação das técnicas de mineração de dados e da análise dos dados, a fim de que estes sejam passíveis de serem compartilhados em um âmbito público para visualização. A fim de melhorar a dinâmica de interação com o usuário, foram disponibilizados filtros dos resultados, de maneira a proporcionar diferentes formas de visualização destes, bem como a possibilidade de criação de diferentes interpretações e pontos de vista sobre os resultados da análise. Esta plataforma foi desenvolvida utilizando as ferramentas apresentadas na seção 3.2.3 Tecnologias/Ferramentas, aplicando-as de forma a seguir o proposto para este trabalho.
- **Disponibilização da plataforma:** após concluída, a plataforma desenvolvida foi hospedada em um servidor e disponibilizada para que possa ser acessada por pessoas que tenham interesse nos resultados da análise, sendo assim aberta ao público em geral de modo que pessoas que queiram detalhes da análise e visualizar os resultados de forma dinâmica possam acessá-la gratuitamente.

4 RESULTADOS E DISCUSSÃO

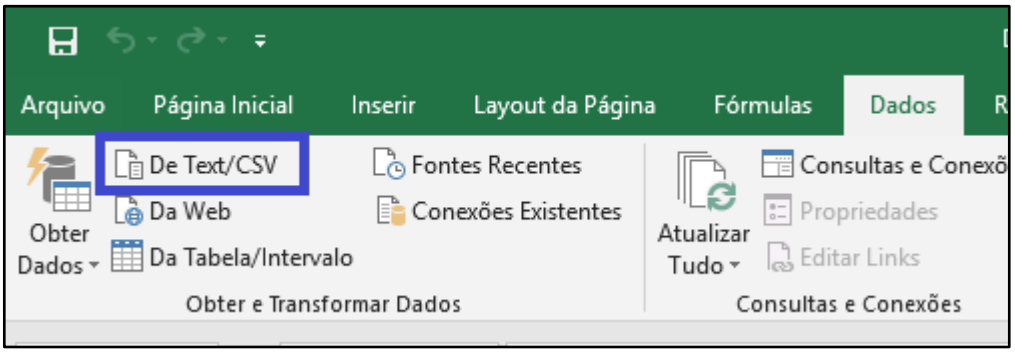
Nesta seção serão descritas as fases do desenvolvimento do trabalho de acordo com o proposto na metodologia. Serão abordadas as etapas de extração, entendimento e transformação dos dados, adequações aplicadas à proposta de acordo com a necessidade, aplicação dos algoritmos de mineração e análise dos resultados.

4.1 EXTRAÇÃO DOS DADOS

As tabelas contendo os dados a serem analisados foram coletadas do website do Inep em formato de tabelas .csv e arquivos textuais .txt, estruturados em formato de microdados. Como estabelecido na proposta deste trabalho foram selecionados os dados referentes aos anos de 2005, 2008, 2011 e 2014, nos quais ocorreram a aplicação das provas do ENADE na área de Computação.

Após a coleta, foi feita a conversão dos arquivos em formato de texto (txt) para arquivos em formato csv, de maneira a facilitar o entendimento de seu conteúdo. Para executar essa tarefa, foi utilizada a ferramenta de importação de arquivos de texto estruturados do Microsoft Excel: “Obter dados de Text/CSV”, destacada em azul na Figura 8.

Figura 8 - Ferramenta de importação de arquivos de texto para CSV



Esta ferramenta realiza a conversão por meio de um delimitador (no caso a vírgula) consegue estruturar os dados textuais em formato de tabelas em formato CSV. Este processo foi aplicado aos arquivos de microdados dos anos de 2011 e 2014 que apresentavam formato de texto.

Com as tabelas no formato adequado, foram aplicados filtros sobre estes para a separação dos cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação. Para isso, foram utilizados os dicionários de variáveis dos dados de cada ano como guia para identificar as colunas que relacionam cada curso. Para o ano de 2005 foi utilizada a coluna `co_grupo`. Para 2008 foram utilizadas as colunas `co_grupo` e

co_subarea, representando a área geral do curso e especificação do curso respectivamente. Para 2011 e 2014, devido à similaridade na estrutura, foi utilizada a coluna co_grupo que, ao contrário do uso dado para o atributo no ano de 2005, informa diretamente o curso do respondente. Com as colunas identificadas e os dados filtrados, estes foram separados em tabelas para cada ano, para auxiliar no entendimento e tratamento individual de forma a permitir a aplicação dos algoritmos de mineração de dados.

Ao fim do processo foi feito o *download* das provas correspondentes aos anos selecionados em formato .pdf para que fosse possível enquadrar as questões nas áreas abordadas de acordo com seu enunciado e opções de respostas. As questões foram enquadradas em suas áreas e armazenadas em uma tabela no formato .xlsx. A Figura 10 representa esta tabela com parte das questões de 2008.

Figura 10 - Tabela de enquadramento das questões por área e ano.

Área da questão	Numero _Curso	Numero	Ano da prova	Curso
Arquitetura de Computadores	1	11	2008	Comp.Específico
Estruturas de dados	2	12	2008	Comp.Específico
Lógica	3	13	2008	Comp.Específico
Estruturas de dados	4	14	2008	Comp.Específico
Redes de Computadores	5	15	2008	Comp.Específico
Engenharia de Software	6	16	2008	Comp.Específico
Lógica	7	17	2008	Comp.Específico
Teorias da Computação	8	18	2008	Comp.Específico
Sistemas Operacionais	9	19	2008	Comp.Específico
Estruturas de dados	10	21	2008	Ciência da Computação
Sistemas Operacionais	10	41	2008	Engenharia da Computação
Engenharia de Software	10	61	2008	Sistemas de Informação
Teorias da Computação	11	22	2008	Ciência da Computação
Redes de Computadores	11	42	2008	Engenharia da Computação
Engenharia de Software	11	62	2008	Sistemas de Informação
Arquitetura de Computadores	12	43	2008	Engenharia da Computação
Banco de dados	12	23	2008	Ciência da Computação
Banco de dados	12	63	2008	Sistemas de Informação
Arquitetura de Computadores	13	24	2008	Ciência da Computação
Banco de dados	13	44	2008	Engenharia da Computação
Engenharia de Software	13	64	2008	Sistemas de Informação

Ao se analisar as provas para os anos de 2008 e 2011 foi observado que a numeração das questões, com exceção das de componente específico, segue em ordem crescente, sem fazer distinção entre as questões específicas para cada curso, a não ser pelo citado no cabeçalho da prova. Já no ano de 2014 foram aplicadas 3 provas completamente distintas uma para cada curso. Como forma de distinguir as áreas das questões, estas foram agrupadas por campo de estudo, número referente ao curso (coluna numero_curso), número referente a prova, ano da prova e curso correspondente.

4.2 ENTENDIMENTO DOS DADOS

Com as tabelas filtradas foi feita uma análise sobre os dados a fim de entendê-los. Nesta etapa de entendimento foram observados os padrões nos microdados e quais colunas das tabelas seriam utilizadas para a obtenção de resultados de acordo os objetivos do trabalho. A partir dessa análise foi verificado que não seria possível incluir neste trabalho os dados do ano de 2005 devido a impossibilidade de distinguir o curso de cada participante que respondeu à prova, tendo em vista que os dados de 2005 estão todos agrupados em apenas uma área denominada “Computação” de forma a englobar todos os cursos existentes da área.

Diante disso foram analisadas as questões das provas do ENADE e seus gabaritos correspondentes nas tabelas de dados para os anos de 2008, 2011 e 2014. A Tabela 4 apresenta a divisão de questões por ano, curso e quantidade de questões de conhecimento geral para acadêmicos da área de Computação (Comp.específico) e conhecimentos que são apenas abordados em disciplinas específicas de cada curso.

Tabela 4 - Divisão de questões por curso e por ano.

Ano	**Curso**	**Número**
2008	Comp.Específico	9
	Ciência da Computação	18
	Engenharia da Computação	18
	Sistemas de Informação	18
2011	Comp.Específico	23
	Ciência da Computação	5
	Engenharia da Computação	5
	Sistemas de Informação	5
2014	Ciência da Computação	27
	Engenharia da Computação	27
	Sistemas de Informação	27

Total Geral	182

4.3 ADEQUAÇÕES DA PROPOSTA

Conforme os conhecimentos adquiridos na etapa de entendimento dos dados foram necessárias adequações na proposta a fim de atingir os objetivos deste trabalho. A principal alteração feita na proposta foi na tabela que contém as áreas para enquadramento das questões. Foi necessário revisar quais matérias seriam agrupadas dentro das áreas de enquadramento de acordo com os temas das questões das provas observadas na Figura 9, seção 4.1. Ao final das análises restaram 12 áreas para enquadramento das questões que se encaixavam nas áreas das questões da prova. A Tabela 5 apresenta as subáreas da Computação que são abrangidas pelas áreas para enquadramento das questões reformuladas.

Tabela 5 - Adequações nas áreas para enquadramento das questões

Grandes áreas	**Subáreas**	**Áreas para enquadramento das questões**
Análise e Desenvolvimento	• Engenharia de Software I • Banco de Dados II • Engenharia de Software II • Modelagem de Sistemas de Informação • Linguagem de Programação para a Web • Gerência de Projetos • Desenvolvimento de Sistemas de Informação • Arquitetura de Software • Qualidade e Auditoria de Software	1. Engenharia de Software 2. Bancos de Dados
Linguagens de Programação	• Algoritmos e Programação I • Algoritmos e Programação II • Linguagem de Programação Orientada a Objetos I • Linguagem de Programação Comercial I • Linguagem de Programação	

	para a Web	
Fundamentos das Ciências exatas	• Fundamentos de Matemática • Cálculo I • Estatística Aplicada • Lógica de Predicados • Matemática Discreta • Geometria Analítica e Álgebra Linear • Cálculo Numérico Computacional	3. Lógica 4. Tópicos avançados de Engenharia
Fundamentos da Computação	• Algoritmos e Programação I • Introdução à Computação • Arquitetura e Organização de Computadores I • Algoritmos e Programação II • Banco de Dados I • Estruturas de Dados I • Estruturas de Dados II • Sistemas Operacionais I	5. Arquitetura de Computadores 6. Estrutura de Dados 7. Sistemas Operacionais
Teorias da Computação	• Fundamentos de Sistemas de Informação • Paradigmas de Linguagens de Programação • Linguagens Formais • Compiladores • Análise de Algoritmos	8. Teorias da Computação

Tecnologias da Computação	• Sistemas de Informação I • Sistemas de Informação II • Interface Homem-Computador • Redes de Computadores I • Inteligência Artificial I • Computação Gráfica • Sistemas Distribuídos • Computação Paralela • Segurança de Sistemas • Redes de Computadores II	9. Sistemas de Informação 10. Redes de Computadores 11. Inteligência Artificial 12. Computação Gráfica

Observando a Tabela 5 é possível notar a ausência de uma área de enquadramento para as subáreas de desenvolvimento de sistemas, que é considerada uma área comum nos cursos da área de Computação em geral. Isso ocorreu, pois durante a análise das questões do ENADE e a classificação das áreas correspondentes, não se identificou um número suficiente de questões para justificar a criação de uma área de enquadramento.

Em conjunto a identificação da área de enquadramento das questões foi verificado se estas estavam aptas para análise de acordo com o gabarito proposto para cada ano nas tabelas de dados do ENADE. Diferentemente do gabarito disposto no site, haviam no gabarito das tabelas certas questões excluídas devido ao seu coeficiente pontobisserial calculado pelo ENADE ser menor que 0.20. Segundo o Relatório Síntese do ENADE[4], as questões necessitam de um poder mínimo de discriminação. Sendo que, o valor do coeficiente pontobisserial serve como índice para indicar se determinada questão da prova está apta a avaliar o desempenho dos alunos, de modo que para ser considerada válida esta deve ter sido acertada mais por alunos com desempenho bom que por alunos com desempenho ruim. Como o objetivo deste trabalho é analisar o desempenho dos estudantes as questões não consideradas aptas foram ignoradas durante o processo de mineração e análise dos dados.

Conforme esta análise das questões aptas realizada pelo Inep para cada prova algumas áreas acabaram por ser ignoradas para cada ano. Isso é possível observar na Figura 11 que apresenta em vermelho as questões que foram ignoradas para os dados de 2011 de acordo com a análise de pontobisserial do Inep.

[4] http://inep.gov.br/relatorios

Figura 11 - Questões ignoradas de acordo com o critério do Inep para os dados de 2011.

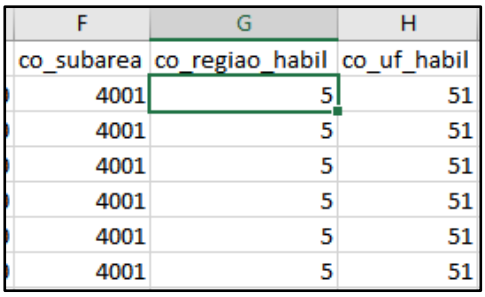
Área da questão	Numero Dados	Numero	Ano da prova	Curso
Arquitetura de Computadores	9	17	2011	Comp.Específico
Inteligência Artificial	27	40	2011	Ciência da Computação
Engenharia de Software	11	19	2011	Comp.Específico
Estruturas de dados	13	21	2011	Comp.Específico
Estruturas de dados	16	24	2011	Comp.Específico
Estruturas de dados	22	30	2011	Comp.Específico
Lógica	1	9	2011	Comp.Específico
Lógica	2	10	2011	Comp.Específico
Lógica	6	14	2011	Comp.Específico
Lógica	12	20	2011	Comp.Específico
Lógica	14	22	2011	Comp.Específico
Lógica	18	26	2011	Comp.Específico
Redes de Computadores	7	15	2011	Comp.Específico
Redes de Computadores	8	16	2011	Comp.Específico
Teorias da Computação	23	36	2011	Ciência da Computação
Teorias da Computação	24	37	2011	Ciência da Computação
Teorias da Computação	25	38	2011	Ciência da Computação
Teorias da Computação	26	39	2011	Ciência da Computação
Sistemas Operacionais	5	13	2011	Comp.Específico
Sistemas Operacionais	10	18	2011	Comp.Específico
Sistemas Operacionais	19	27	2011	Comp.Específico
Sistemas Operacionais	21	29	2011	Comp.Específico
Teorias da Computação	3	11	2011	Comp.Específico
Teorias da Computação	4	12	2011	Comp.Específico
Teorias da Computação	15	23	2011	Comp.Específico
Teorias da Computação	17	25	2011	Comp.Específico
Teorias da Computação	20	28	2011	Comp.Específico

4.4 TRANSFORMAÇÃO E ADEQUAÇÃO DOS DADOS PARA ANÁLISE

A partir do entendimento dos dados e das adequações na proposta foi feita a transformação e adequação dos dados. Inicialmente os dados foram filtrados de acordo com os códigos referente à área de Computação na coluna `co_grupo` e à presença do estudante na prova na coluna `tp_press` com valor "555" indicando sua presença na prova. Os dados resultantes da filtragem foram armazenando uma nova tabela para que fossem submetidos ao processo de padronização.

Com a separação das respostas da área de Computação os dados foram submetidos a uma padronização e à conversão de valores numéricos em textuais de modo que fosse possível a aplicação dos algoritmos de mineração. Para a execução de processos de padronização e conversão de valores, foi utilizada a funcionalidade '*Localizar e Substituir'* do Microsoft Excel. A partir disso, determinados valores, como os códigos dos estados e os códigos das regiões em que o aluno reside foram convertidos para suas respectivas siglas dos estados e nomes das regiões. A Figura 12 apresenta um exemplo dos dados antes da transformação, seguida da Figura 13 exibindo um exemplo dos dados já transformados.

Figura 12 - Colunas co_subarea, co_uf_habil e cod_regiao_habil nos dados de 2008.



F	G	H
co_subarea	co_regiao_habil	co_uf_habil
4001	5	51
4001	5	51
4001	5	51
4001	5	51
4001	5	51
4001	5	51

Figura 13 - Colunas co_subarea, co_uf_habil e cod_regiao_habil contendo a transformação dos dados de 2008.

F	G	H
co_subarea	co_regiao_habil	co_uf_habil
Ciência da Computacao	Sudeste	MG
Ciência da Computacao	Sudeste	MG
Ciência da Computacao	Sudeste	MG
Ciência da Computacao	Sudeste	MG
Ciência da Computacao	Sudeste	MG
Ciência da Computacao	Sudeste	MG

Durante o decorrer do desenvolvimento desta etapa foi encontrado um obstáculo ao se comparar os gabaritos das provas com as respostas dos alunos. Para analisar o desempenho dos alunos de acordo com o campo de estudo das questões, foi necessário que as respostas fossem analisadas individualmente, algo que, com o formato original dos dados, era inviável devido ao gabarito da prova e as respostas às questões estarem agrupados em vetores. A Figura 14 apresenta um exemplo do gabarito e das questões da prova de 2011 sem a transformação dos dados.

Figura 14 - Gabarito e respostas dos alunos no ano de 2011.

Y	AB
DS_VT_GAB_OCE_FIN (A)	DS_VT_ESC_OCE (B)
XEXXZEDAXBEDACXXEXEBACNNNNNNNNNNNNNNNAXCBA	DBAECBECEACBEDACEDCBEA.....DEADC
XEXXZEDAXBEDACXXEXEBACNNNNNNNNNNNNNNNAXCBA	CCECCBECCDECCACDEBCCDB.....CBCDB
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	CEDCCCABBEDCCBBBECDBCE.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	BCBCBAEDCBACCCCBCBABAB.....BABAC.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	AECCADBABEDBACBDEDBDBA.....BEAEC.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	BEECDCDABDEACABBDACBEC.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	ADEDEBCECAECCCCBDCCCEB.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	DECBBDEECDAADABCABDDCA.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	AEABBEAAADCCCADBAADDCD.....
DECBZEDAXBEDACXXEDEBACNNNNNDDXDXNNNNNNNNNN	EECDAACEBCCAAAEDDCEDEC.....

A esquerda (A) da Figura 14 tem-se o gabarito da prova contendo as letras de A a E para identificação das respostas corretas, X para indicar uma resposta nula e N para identificar que a questão não condiz ao curso do aluno. Observa-se também que há dois gabaritos distintos, os quais correspondem às provas de Ciência da Computação, a partir da terceira linha, e Sistemas de Informação que seguem o mesmo padrão de apresentação dos dados.

A direita (B) da Figura 14 tem-se as respostas de alunos, contendo letras de A a E como assertivas assinaladas e um ponto (‘.’) a fim de indicar que ou o aluno não respondeu à questão ou esta não necessitava de sua resposta. É válido ressaltar que este problema não remete apenas ao ano de 2011, mas também aos anos de 2008 e 2014 que apresentam os formatos semelhantes de apresentação do gabarito e das respostas por meio de vetores em texto.

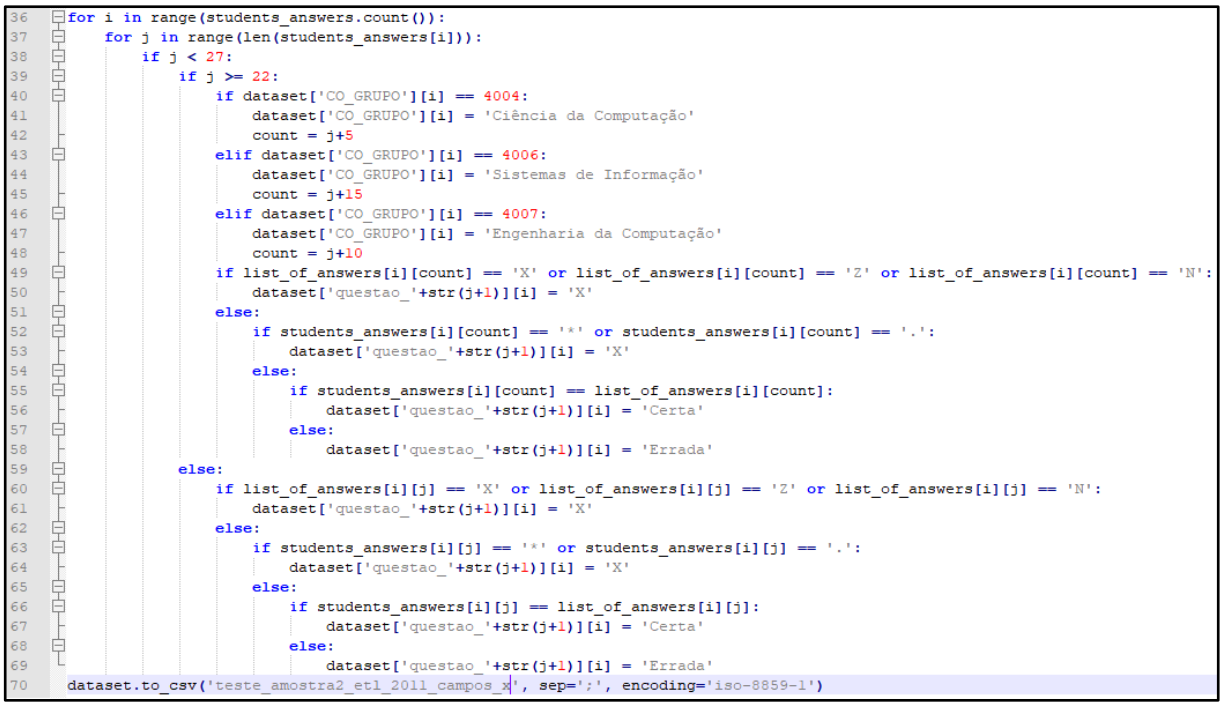
Com o objetivo de contornar o problema foram desenvolvidos três algoritmos em Python para cada ano de prova, estes executaram uma comparação entre o gabarito e as respostas às questões para cada curso de acordo com o formato do ano em questão. Os algoritmos criados utilizam a biblioteca Pandas para leitura e manipulação de arquivos .csv e demais formatos tabulares. A princípio foram criadas colunas de acordo com o total de questões do gabarito (totalizando 27), nessas colunas foram armazenados os resultados das comparações das respostas dos alunos e do gabarito. A Figura 15 exibe a parte do algoritmo responsável pela criação das colunas.

Figura 15 - Criação de colunas no algoritmo de comparação para o ano de 2011

```
19  for i in range(len(list_of_answers[1])):
20      if i < 27:
21          dataset['questao_'+str(i+1)] = ''
```

Para a criação das colunas foi feito um laço de repetição que executa enquanto o valor do contador fosse menor que 27. Enquanto essa condição fosse atendida, em cada iteração, uma coluna “questão” foi adicionada sendo seu cabeçalho uma concatenação do número do contador acrescido de 1, sendo assim criadas as colunas: `questao_1`, `questao_2` ... `questao_27`. Com as colunas definidas foi feita a comparação entre o gabarito e a resposta do estudante. Nesse processo foi atribuído o valor “Certa” caso a resposta estivesse de acordo ao gabarito, “Errada“ caso estivesse divergente e “X” caso o aluno não tenha respondido à questão ou sua resposta não tenha sido validada. A Figura 16 apresenta a parte do algoritmo aplicado sobre os dados de 2011.

Figura 16 - Algoritmo de comparação para os dados de 2011.



```
36  for i in range(students_answers.count()):
37      for j in range(len(students_answers[i])):
38          if j < 27:
39              if j >= 22:
40                  if dataset['CO_GRUPO'][i] == 4004:
41                      dataset['CO_GRUPO'][i] = 'Ciência da Computação'
42                      count = j+5
43                  elif dataset['CO_GRUPO'][i] == 4006:
44                      dataset['CO_GRUPO'][i] = 'Sistemas de Informação'
45                      count = j+15
46                  elif dataset['CO_GRUPO'][i] == 4007:
47                      dataset['CO_GRUPO'][i] = 'Engenharia da Computação'
48                      count = j+10
49                  if list_of_answers[i][count] == 'X' or list_of_answers[i][count] == 'Z' or list_of_answers[i][count] == 'N':
50                      dataset['questao_'+str(j+1)][i] = 'X'
51                  else:
52                      if students_answers[i][count] == '*' or students_answers[i][count] == '.':
53                          dataset['questao_'+str(j+1)][i] = 'X'
54                      else:
55                          if students_answers[i][count] == list_of_answers[i][count]:
56                              dataset['questao_'+str(j+1)][i] = 'Certa'
57                          else:
58                              dataset['questao_'+str(j+1)][i] = 'Errada'
59              else:
60                  if list_of_answers[i][j] == 'X' or list_of_answers[i][j] == 'Z' or list_of_answers[i][j] == 'N':
61                      dataset['questao_'+str(j+1)][i] = 'X'
62                  else:
63                      if students_answers[i][j] == '*' or students_answers[i][j] == '.':
64                          dataset['questao_'+str(j+1)][i] = 'X'
65                      else:
66                          if students_answers[i][j] == list_of_answers[i][j]:
67                              dataset['questao_'+str(j+1)][i] = 'Certa'
68                          else:
69                              dataset['questao_'+str(j+1)][i] = 'Errada'
70  dataset.to_csv('teste_amostra2_etl_2011_campos_x', sep=';', encoding='iso-8859-1')
```

Como observado na Figura 16 o algoritmo percorre a tabela como uma matriz sendo *i* um contador referente às linhas da tabela e *j* referente às posições dos vetores. Na linha 36 do código é possível observar um condicional que verifica se o número de colunas excede às 27 questões para as provas de cada curso. Se o valor do contador fosse menor ele iria verificar se o valor da posição do vetor era menor que 22, a fim de identificar se deveria fazer a comparação para as questões de componente específico onde se comparava o vetor do gabarito na linha *i* em sua posição *j*, caso este fosse igual a X, Z ou N era atribuído o valor X à coluna da questão indicando que a questão foi anulada ou não correspondia ao curso do respondente. Se este não se enquadrasse em nenhum dos casos citados, seria feita a verificação no vetor do aluno para a posição *j*, se o valor for um asterisco (*) ou um ponto(.) seria atribuído o valor X à coluna indicando que o aluno não respondeu ou teve sua resposta inválida. Ao passar pelas verificações anteriores foi feita a comparação do vetor do gabarito na linha *i* na posição *j* com o vetor de respostas do aluno na mesma linha e posição. Se a resposta do aluno estivesse de acordo com o gabarito era atribuído o valor “Certa” e caso não condiga com o gabarito era adicionado o valor “Errada” na coluna da questão referente ao resultado da comparação.

No caso do valor do contador fosse maior que 22 foi feita uma verificação na coluna `co_grupo` para identificar qual o curso do aluno, a partir do curso identificado este valor foi substituído pelo título do curso, bem como, foi adicionado um valor ao contador que

corresponde à posição das respostas às perguntas do curso tanto no vetor do gabarito como no vetor de respostas do estudante. Com o curso definido, foram feitas as validações e a comparação dos campos do vetor do gabarito com o vetor das respostas do estudante seguindo o mesmo padrão abordado anteriormente. Ao fim da execução do algoritmo foi gerada uma nova tabela no formato .csv contendo os resultados da comparação para cada questão de cada curso, bem como o título do curso do respondente na coluna `co_grupo`.

Para testar o resultado do algoritmo foi criada uma tabela com amostra de 20 linhas extraídas da tabela de respostas do ENADE filtradas pela área da Computação, com o resultado do teste foram feitos ajustes nos condicionais do algoritmo a fim de se obter o resultado esperado. A Tabela 6 apresenta a porcentagem de respostas referentes à área de Computação por região.

Tabela 6 - Número e porcentagem de respostas por regiões.

Ano	**Total**	**Região**	**Número de respostas**	**Porcentagem sobre o total**
2008	37.152	Norte	1764	4,75%
		Nordeste	4694	12,63%
		Centro-Oeste	3565	9,60%
		Sudeste	19896	53,55%
		Sul	7235	19,47%
2011	21.913	Norte	1017	4,64%
		Nordeste	3379	15,42%
		Centro-Oeste	1860	8,49%
		Sudeste	11814	53,91%
		Sul	3843	17,54%
2014	24.076	Norte	1426	5,92%
		Nordeste	4073	16,92%

		Centro-Oeste	1975	8,20%
		Sudeste	12270	50,96%
		Sul	4332	17,99%

Essa divisão foi necessária devido à grande quantidade de respostas das regiões Sul e Sudeste, que como é observado na Tabela 6, juntas ocupam, para todos os anos, aproximadamente 70% do total de respostas, sendo em média 52% do total resultados da região Sudeste. Caso os dados fossem analisados todos juntos, poderiam haver dificuldades de conseguir resultados suficientes para verificar o desempenho dos acadêmicos das regiões Norte, Nordeste e Centro-Oeste.

4.5 APLICAÇÃO DOS ALGORITMOS DE MINERAÇÃO DE DADOS

Com a finalidade de obter resultados que estivessem de acordo com a proposta, foram testados diversos algoritmos de classificação, clusterização e associação disponibilizados na plataforma Weka. Para observar e analisar os possíveis resultados destes algoritmos foram criadas tabelas contendo amostras dos dados. Dentro do ambiente do Weka é possível filtrar quais colunas serão utilizadas na execução dos algoritmos. Para os testes foram selecionadas as colunas `co_regiao`, `questao_13`, `questao_19`, `questao_23`, `questao_25` e `questao_27`.

A partir das informações resultantes dos testes executados, foi selecionado o algoritmo de clusterização Simple KMeans para a aplicação nos dados tratados devido ao fato de seus resultados atenderem adequadamente aos objetivos do trabalho. Após definir o algoritmo a ser aplicado, foram feitos testes para escolher a melhor forma de explorador os resultados. Devido ao requisito do algoritmo em definir um número k de grupos a serem gerados foi padronizado o total de k=20, de maneira que com a execução do algoritmo pudessem ser gerados k ou menos grupos. Este número foi definido com base em testes dos resultados com diversos números para k, sendo que o valor de 20 grupos foi o que mais se adequou às necessidades e objetivos da proposta. A Figura 17 apresenta um exemplo de resultado da aplicação do algoritmo Simple KMeans para a amostra dos dados de 2008.

Figura 17 - Exemplo de resultado do algoritmo Simple KMeans.

```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 728.0

Initial starting points (random):

Cluster 0: Sudeste,Certa,Errada,Errada,Errada,Certa
Cluster 1: Sudeste,Errada,Errada,Certa,Errada,Certa
Cluster 2: Sudeste,Errada,Errada,Errada,Errada,Certa
Cluster 3: Sudeste,Errada,Errada,Errada,Errada,Errada
Cluster 4: Sudeste,Errada,Errada,Errada,Certa,Certa
Cluster 5: Sudeste,Certa,Certa,Errada,Errada,Errada
Cluster 6: Sudeste,Errada,Certa,Errada,Errada,Errada
Cluster 7: Sudeste,Errada,Errada,Errada,Certa,Errada
Cluster 8: Sudeste,Errada,Errada,Certa,Errada,Errada
Cluster 9: Sudeste,Errada,Errada,Certa,Certa,Errada
Cluster 10: Sudeste,Errada,Certa,Certa,Errada,Errada
Cluster 11: Sudeste,Errada,Certa,Certa,Certa,Errada
Cluster 12: Sudeste,Errada,Certa,Errada,Certa,Errada
Cluster 13: Sudeste,Certa,Errada,Certa,Certa,Errada
Cluster 14: Sudeste,Certa,Errada,Certa,Errada,Errada
Cluster 15: Sudeste,Certa,Errada,Certa,Errada,Certa
Cluster 16: Sudeste,Errada,Certa,Certa,Errada,Certa
Cluster 17: Sudeste,Certa,Certa,Certa,Certa,Errada
Cluster 18: Sudeste,Errada,Errada,X,Errada,Errada
Cluster 19: Sudeste,Errada,Errada,Certa,Certa,Certa
```

Com esta configuração o algoritmo foi aplicado repetidamente para todo o conjunto de dados preparados. Os resultados desse trabalho serão detalhados com base no seguinte filtro: dados tratados do curso de CC do ano de 2011 para as questões da área de Lógica, que são: 2, 6, 12 14. As colunas selecionadas para a clusterização foram: região e questões referentes aos campos de estudo, seguindo a mesma estrutura do teste apresentado na Figura 15. Os resultados obtidos desta aplicação serão melhor detalhados no tópico 4.6.

4.6 ANÁLISE E VALIDAÇÃO DOS RESULTADOS:

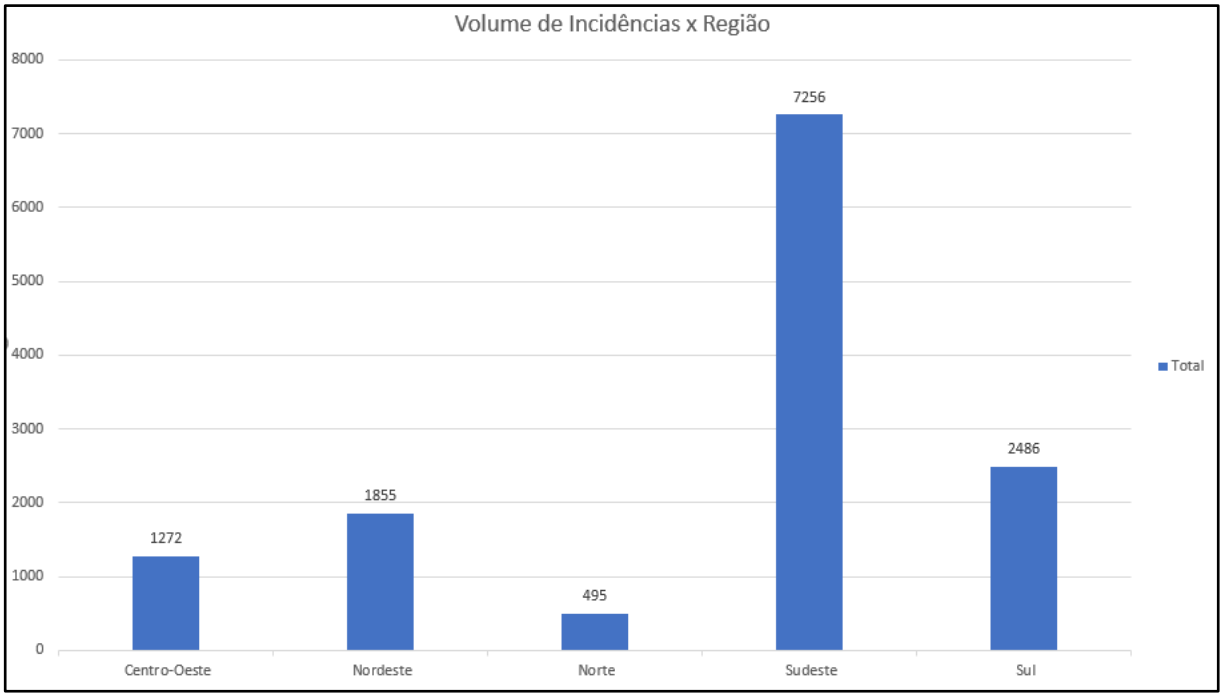
Para a análise do resultado foi feita a divisão dos resultados por curso e por região do Brasil, com isso, os dados foram separados em tabelas específicas para os cursos de Ciência da Computação, Sistemas de Informação e Engenharia da Computação, sendo também separados em tabelas específicas para cada região para esses cursos. A Figura 18 apresenta os resultados obtidos da aplicação do algoritmo sobre as respostas dos acadêmicos do curso de Ciências da Computação da região Norte.

Figura 18 - Resultados da clusterização das questões da área de Lógica do ano de 2011 da região Norte no curso de Ciências da Computação.

area_enquadramento	ano	regiao	volume_incidencias_porcentagem	qtd_questoes	qtd_certas	qtd_erradas	qtd_branco	porcentagem_certas	porcentagem_erradas	porcentagem_branco
Lógica	2011	Norte	0 38 (8%)	6	1	5	0	16,67%	83,33%	0,00%
Lógica	2011	Norte	1 63 (13%)	6	3	3	0	50,00%	50,00%	0,00%
Lógica	2011	Norte	2 71 (14%)	6	0	6	0	0,00%	100,00%	0,00%
Lógica	2011	Norte	3 27 (6%)	6	1	5	0	16,67%	83,33%	0,00%
Lógica	2011	Norte	4 15 (3%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	5 56 (11%)	6	1	5	0	16,67%	83,33%	0,00%
Lógica	2011	Norte	8 45 (9%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	9 18 (4%)	6	1	5	0	16,67%	83,33%	0,00%
Lógica	2011	Norte	10 15 (3%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	13 26 (5%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	15 18 (4%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	16 16 (3%)	6	2	4	0	33,33%	66,67%	0,00%
Lógica	2011	Norte	17 24 (5%)	6	1	5	0	16,67%	83,33%	0,00%

A Figura 18 apresenta os clusters selecionados a partir da aplicação do algoritmo de mineração de dados sobre as respostas às questões de Lógica de CC do ano de 2011 da região do Norte em tabela em um formato padronizado. A partir destes resultados foram gerados gráficos que representassem a taxa de questões certas em cada cluster em função do número de incidências ocorridas naquele grupo. Deste modo é possível observar a taxa de questões em que foram obtidas respostas corretas e erradas por região. A Figura 19 apresenta um gráfico do volume de incidências (quantidade de dados) por região, resultantes da aplicação do algoritmo sobre os dados de 2011.

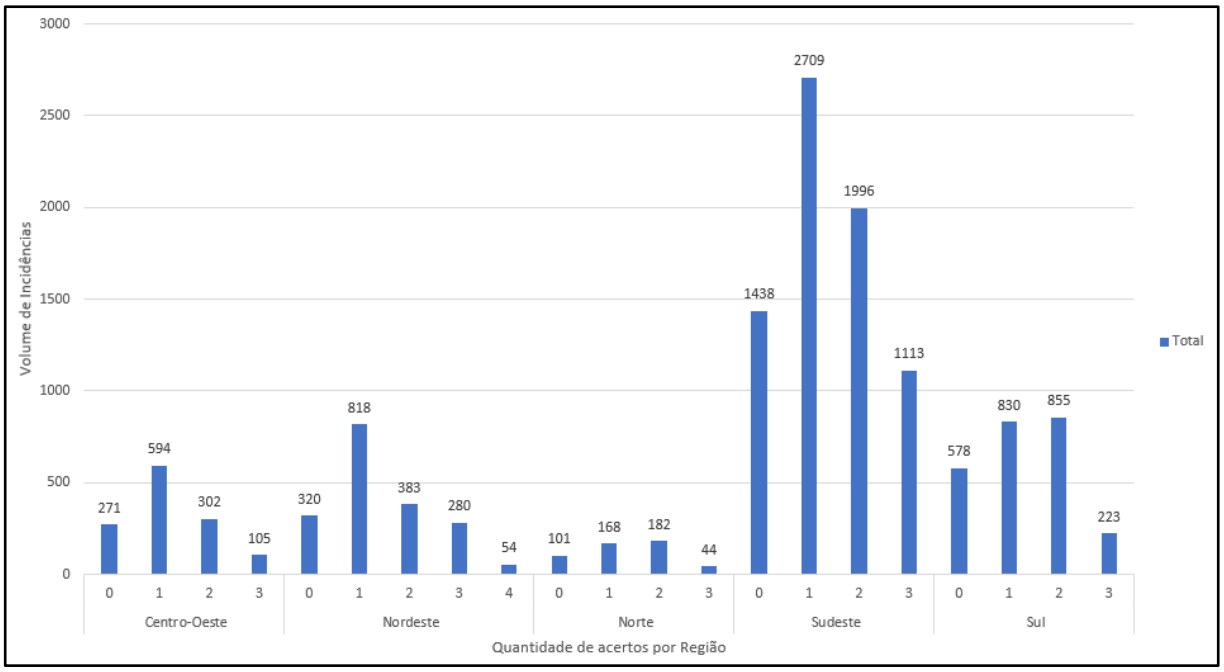
Figura 19 - Gráfico de volume de Incidências por região dos resultados de 2011.



Para interpretar estes resultados, foram excluídos grupos que continham um volume de incidências que correspondessem a menos de 3% do volume de dados originais, de maneira a

ignorar possíveis clusters que não obtiveram valores relevantes e que não seriam interessantes para a análise devido ao seu pouquíssimo volume. A Figura 20 apresenta o gráfico resultante da aplicação do algoritmo para as cinco regiões sobre as respostas com base nos filtros citados acima.

Figura 20 - Gráfico de volume de incidências por acertos para cada região em Lógica no ano de 2011.



O gráfico disposto na Figura 20 apresenta o volume de incidências (eixo Y) em função da quantidade de respostas corretas dos discentes de Ciência da Computação (eixo X) nas questões da área de Lógica para cada região em 2011.

Ao realizar uma análise estatística sobre o desempenho geral dos discentes nas questões da área, é possível considerar que os discentes da região Sudeste obtiveram a maior taxa de acerto nas questões da área de Arquitetura de Computadores, tendo um total 5818 (aproximadamente 80,18%) incidências, das 7256 incidências dos clusters, considerando que os discentes acertaram ao menos uma questão. Seguida da região Nordeste, que apesar de um menor volume de respostas em comparação a região Sul teve um total de 1481 (cerca de 79,83%) com base nos mesmos dados. Com a menor taxa de acertos tem-se a região Sul com um total de 1908 (em torno de 76,7%) em um total de 2486 respostas.

Com os dados organizados desta forma, foi possível replicar a análise para os demais anos e áreas de conhecimento da Computação. A plataforma de visualização dos resultados será apresentada na seção 4.7.

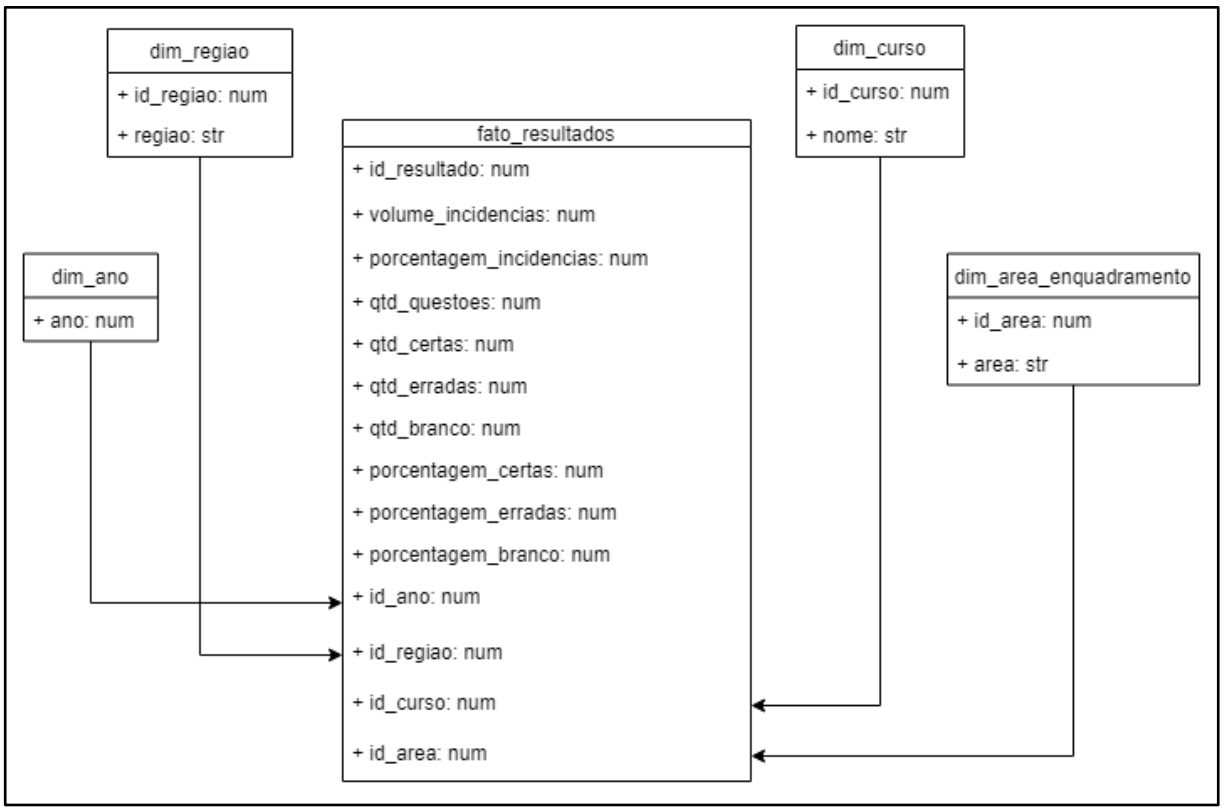
4.7 ELABORAÇÃO DA PLATAFORMA PARA APRESENTAÇÃO DOS RESULTADOS

A plataforma EnadeDM foi inicialmente idealizada como uma forma de visualização dos resultados da aplicação dos algoritmos de mineração de dados em formato de gráfico. Os tópicos desta seção abordam os processos do desenvolvimento da plataforma abrangendo desde a modelagem do banco de dados até a estruturação de sua parte visual.

4.7.1 Modelagem do BD

A primeira etapa, envolveu a estruturação de como os resultados seriam armazenados para disponibilização dentro da plataforma. Para isso foi criado um diagrama de classes que representasse a estrutura da base de dados a ser preenchida, apresentando as tabelas e seus relacionamentos. A Figura 21 representa este modelo.

Figura 21 - Diagrama de classes da estrutura do BD do EnadeDM



O diagrama apresentado na Figura 21 foi estruturado a partir do conceito de utilização de Data Warehouse para armazenamento de dados para análises. Este modelo foi desenvolvido pensando nas possibilidades de adição de novos dados resultantes da aplicação de técnicas de mineração de dados sobre os dados do ENADE em diferentes áreas. Além

disso, o modelo proporciona um formato de busca aos dados de modo mais interativo por meio de filtros elaborados no *Front-end*.

4.7.2 Inserção dos resultados no BD.

A partir do modelo foi criada a base de dados da API a ser consumida pela plataforma utilizando Django Rest Framework. Para seguir o modelo de DW foram associados números para cada região, área de enquadramento e curso. Isto pode ser observado na Figura 21 em que tabela `Fato` armazena as informações responsáveis pelas respostas e os ID's associados às tabelas dimensões dos campos que caracterizam aquela resposta, no caso a região, curso, ano e área de enquadramento. Diante disso, foi criado um documento em formato de tabela .xlsx que descreve os números associados às informações das tabelas `dimensão`. A Figura 22 apresenta este documento.

Figura 22 - Documentação de identificadores do BD do EnadeDM.

Admin	
User	Senha
admin	******

Dim_area_enquadramento	
ID	Área
1	Engenharia de Software
2	Bancos de Dados
3	Lógica
4	Tópicos Avançados de Engenharia
5	Arquitetura de Computadores
6	Estruturas de Dados
7	Sistemas Operacionais
8	Teorias da Computação
9	Sistemas de Informação
10	Redes de Computadores
11	Inteligência Artificial
12	Computação Gráfica

Região	
ID	Região
1	Norte
2	Nordeste
3	Centro-Oeste
4	Sudeste
5	Sul

Curso	
ID	Curso
1	Ciência da Computação
2	Sistemas de Informação
3	Engenharia da Computação

Os identificadores apresentados na Figura 22 foram cadastrados diretamente no BD utilizando o Django Admin, de modo que já estivessem disponíveis para que na hora da inserção não fossem gerados conflitos de relacionamentos entre as tabelas. Para a inserção dos resultados no banco, os resultados obtidos foram agrupados em tabelas no formato .csv, combinando os resultados da clusterização para cada área de enquadramento das questões e

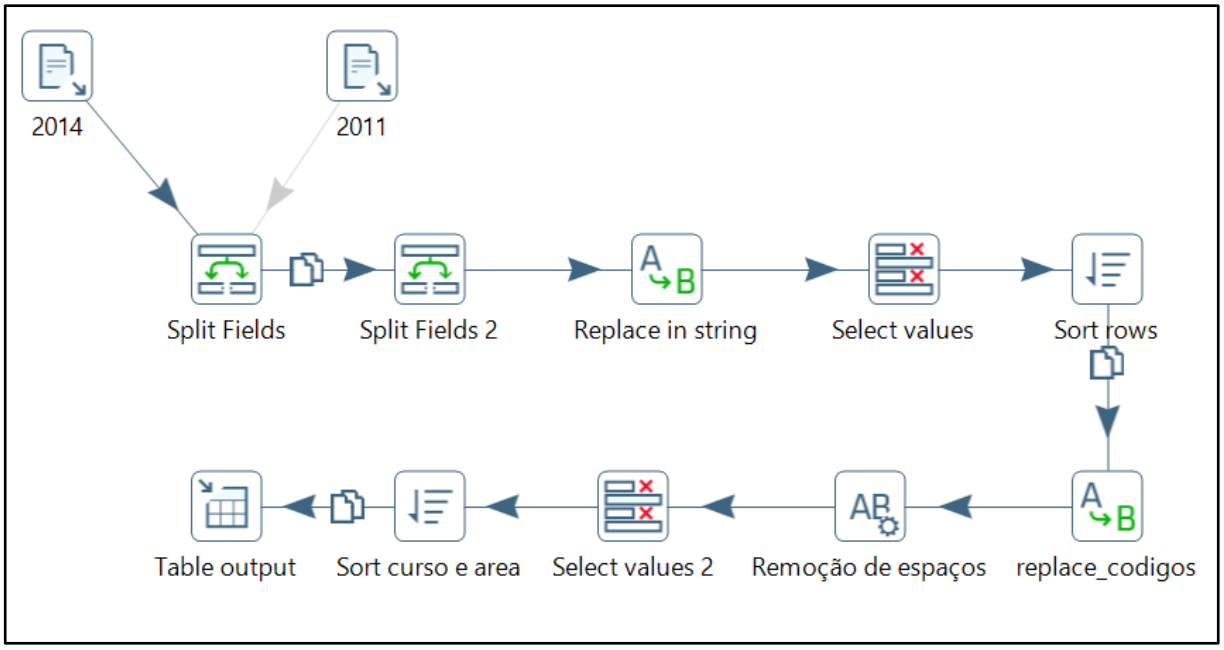
para cada região em tabelas únicas para cada ano. A Figura 23 apresenta um fragmento da tabela resultante desse processo para o ano de 2014.

Figura 23 - Tabela de resultados de 2014 para inserção no banco de dados.

area_enquadramento	curso	ano	regiao	volume_incidencias	qtd_questoes	qtd_certas	qtd_erradas	qtd_branco_	porcentagem_certas	porcentagem_erradas	porcentagem_branco
Teorias da Computação	Ciência da Computação	2014	Sudeste	11 287 (7%)	7	1	6	0	14,29	85,71	0
Teorias da Computação	Ciência da Computação	2014	Sudeste	12 176 (4%)	7	5	2	0	71,43	28,57	0
Teorias da Computação	Ciência da Computação	2014	Sudeste	14 171 (4%)	7	3	4	0	42,86	57,14	0
Teorias da Computação	Ciência da Computação	2014	Sudeste	16 127 (3%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	0 162 (10%)	7	3	4	0	42,86	57,14	0
Teorias da Computação	Ciência da Computação	2014	Sul	1 218 (13%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	2 87 (5%)	7	5	2	0	71,43	28,57	0
Teorias da Computação	Ciência da Computação	2014	Sul	3 74 (4%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	4 83 (5%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	5 144 (9%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	6 61 (4%)	7	3	4	0	42,86	57,14	0
Teorias da Computação	Ciência da Computação	2014	Sul	8 133 (8%)	7	1	6	0	14,29	85,71	0
Teorias da Computação	Ciência da Computação	2014	Sul	10 85 (5%)	7	4	3	0	57,14	42,86	0
Teorias da Computação	Ciência da Computação	2014	Sul	12 46 (3%)	7	3	4	0	42,86	57,14	0
Teorias da Computação	Ciência da Computação	2014	Sul	14 123 (7%)	7	1	6	0	14,29	85,71	0
Teorias da Computação	Ciência da Computação	2014	Sul	15 45 (3%)	7	2	5	0	28,57	71,43	0
Teorias da Computação	Ciência da Computação	2014	Sul	16 208 (13%)	7	0	7	0	0	100	0
Arquitetura de Computadores	Engenharia da Computação	2014	Sudeste	0 92 (7%)	3	2	1	0	66,67	33,33	0
Arquitetura de Computadores	Engenharia da Computação	2014	Norte	0 53 (31%)	3	1	2	0	33,33	66,67	0
Arquitetura de Computadores	Engenharia da Computação	2014	Sul	0 103 (26%)	3	1	2	0	33,33	66,67	0
Arquitetura de Computadores	Engenharia da Computação	2014	Nordeste	0 146 (33%)	3	0	3	0	0	100	0
Arquitetura de Computadores	Engenharia da Computação	2014	Sul	1 28 (7%)	3	1	2	0	33,33	66,67	0
Arquitetura de Computadores	Engenharia da Computação	2014	Norte	1 43 (25%)	3	0	3	0	0	100	0
Arquitetura de Computadores	Engenharia da Computação	2014	Sudeste	1 460 (33%)	3	0	3	0	0	100	0
Arquitetura de Computadores	Engenharia da Computação	2014	Nordeste	1 96 (22%)	3	1	2	0	33,33	66,67	0
Arquitetura de Computadores	Engenharia da Computação	2014	Sudeste	2 131 (9%)	3	1	2	0	33,33	66,67	0

De forma a adequar os dados da tabela para os padrões do banco de dados e inseri-los foi utilizado a ferramenta para ETL PDI. Diante disso, foi desenvolvido um modelo de transformação e padronização na ferramenta que executa a conversão dos valores de texto das colunas: `area_enquadramento`, `curso` e `região`, nos respectivos ID's já cadastrados na base de dados. A Figura 24 apresenta o modelo criado na ferramenta para o tratamento e carga das informações na base de dados.

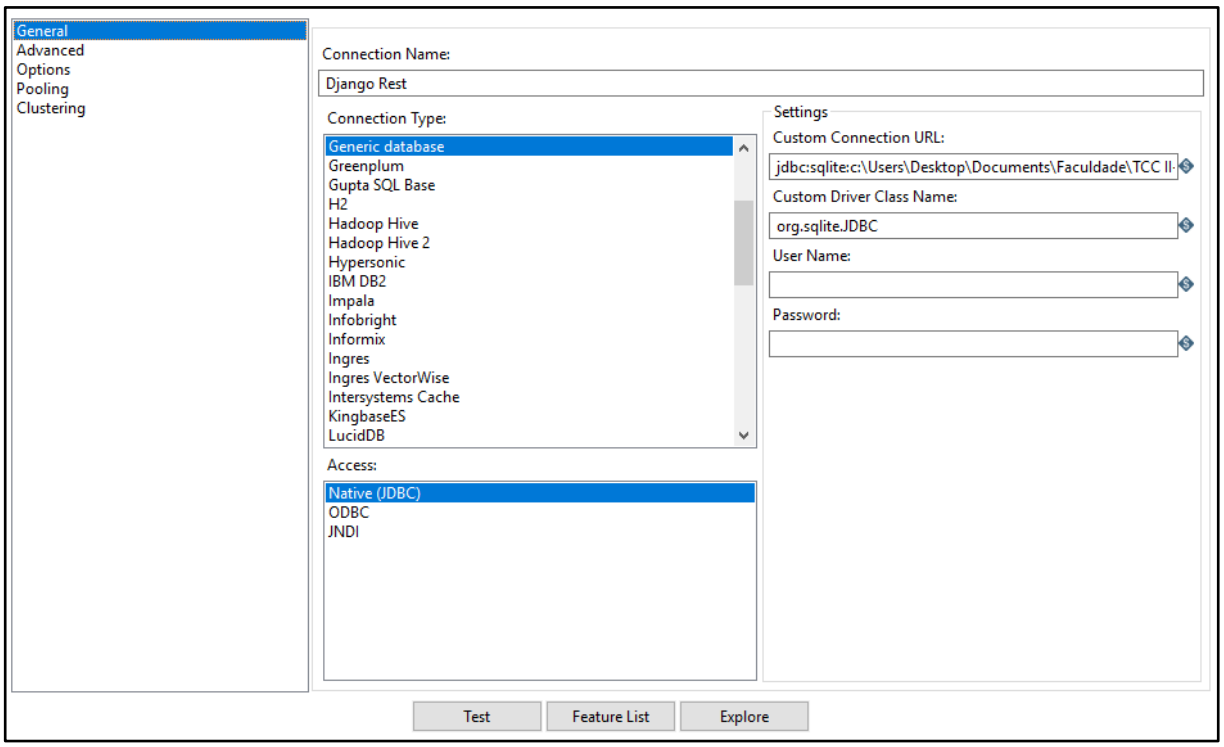
Figura 24 - Modelo de padronização, transformação e carga dos dados de 2011 e 2014 no BD.



No modelo apresentado na Figura 24 também foi feito o tratamento dos valores da coluna `volume_incidências`, separando-os em duas colunas `volume_incidencias` e `porcentagem_incidencias`, de maneira que os valores fossem separados para a apresentação na plataforma. Além disso, foi feita a transformação dos dados das colunas que representam as porcentagens, que para armazenamento na base de dados em formato de número (Number) foi removido o símbolo "%" do seu final. O passo seguinte foi remover possíveis espaços em branco dos dados que pudessem caracterizá-los como tipo String e não como número.

Para a carga dos dados padronizados no BD foi necessária a configuração das informações do banco na ferramenta, de modo que todo o processo é feito continuamente sem necessidade de executar os passos manualmente, desta forma automatizando o processo de transformação e carga. A Figura 25 apresenta a configuração necessária para conexão com o BD da plataforma.

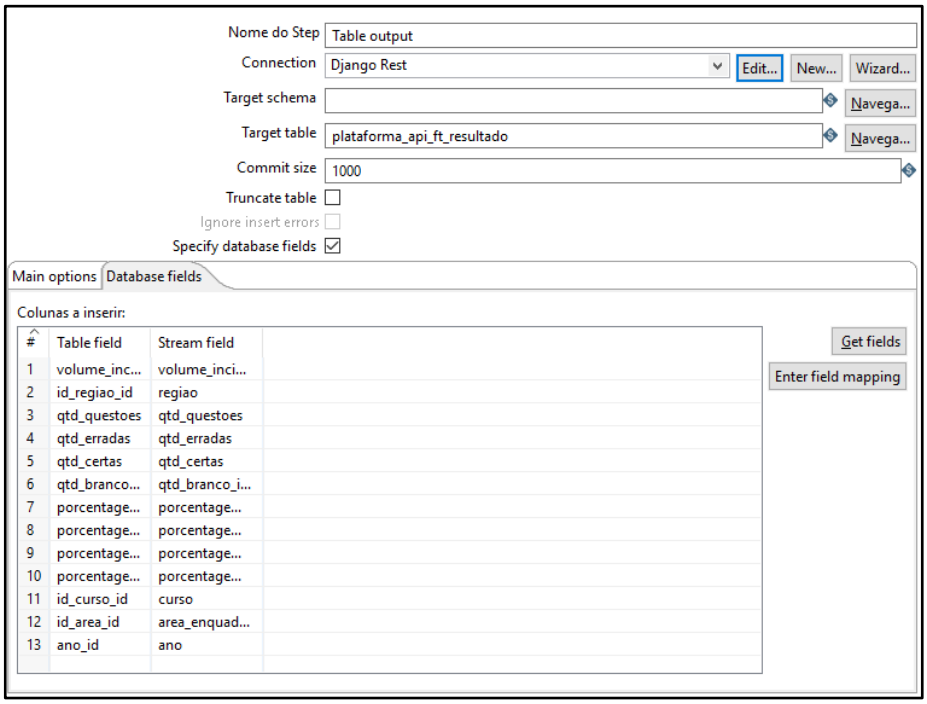
Figura 25 - Configuração do SQLite no Pentaho Data Integration.



Devido à base de dados da API ser em SQLite, foi necessária a configuração desta na ferramenta como uma base de dados genérica com uma conexão customizada para o endereço do arquivo no computador. Além disso a conexão teve de ser instanciada como nativa JDBC e com o nome da classe do driver customizada para o padrão do SQLite. A fim de que garantir que os dados oriundos das tabelas .csv fossem inseridos nos campos corretamente, foram

relacionados os campos no PDI de modo que cada campo resultante do ETL fosse relacionado ao seu equivalente no BD. A Figura 26 apresenta estes campos.

Figura 26 - Relacionamento das colunas das tabelas de dados com as colunas do BD.



Observando a Figura 26, é possível verificar que os dados foram inseridos na tabela alvo (Target table) `ft_resultado`, de modo a seguir os padrões estipulados para DWs em que as informações principais são armazenadas em uma tabela fato. Além disso é feito o relacionamento entre as colunas resultantes do ETL (*Stream Field*) e as tabelas contidas na base de dados (*Table Field*). Com o modelo estruturado foram executados testes antes de realizar a inserção dos dados diretamente na base de dados, que pode ser observado através dos resultados gerados pela função *Preview* do PDI. Ao validar o modelo, este foi executado de modo que os dados fossem inseridos dentro do BD para que assim estes estivessem prontos para serem utilizados por meio da API.

4.7.3 Desenvolvimento da API.

O desenvolvimento da API para a plataforma seguiu o fluxo de desenvolvimento indicado pela documentação do Django Rest Framework, sendo: Criar o modelo, serializar, criar as Views e URLs. A partir do modelo já criado na etapa de inserção dos dados, foram configurados os *Serializers*[5] no arquivo `serializers.py`, que segundo a documentação do

[5] https://www.django-rest-framework.org/api-guide/serializers/

Django Rest Framework, a partir do modelo de dados converte dados complexos em tipos de dados nativos do Python que podem ser facilmente convertidos em formatos JSON e XML. define os campos que serão serializados para retorno às requisições. A Figura 27 apresenta o *Serializer* criado para retorno dos resultados da mineração.

Figura 27 - Serializer dos resultados

```
4   class ResultadoSerializer(serializers.Serializer):
5     ano = serializers.SlugRelatedField(slug_field = 'ano', read_only=True)
6     id_area = serializers.SlugRelatedField(slug_field = 'area', read_only=True)
7     id_curso = serializers.SlugRelatedField(slug_field = 'curso', read_only=True)
8     id_regiao = serializers.SlugRelatedField(slug_field = 'regiao', read_only=True)
9     volume_incidencias = serializers.IntegerField()
10    porcentagem_incidencias = serializers.FloatField()
11    qtd_questoes = serializers.IntegerField()
12    qtd_certas = serializers.IntegerField()
13    qtd_erradas = serializers.IntegerField()
14    qtd_branco_invalidas = serializers.IntegerField()
15    porcentagem_certas = serializers.FloatField()
16    porcentagem_erradas = serializers.FloatField()
17    porcentagem_branco_invalida = serializers.FloatField()
18    class Meta:
19      model = Ft_resultado
20      fields = ('volume_incidencias', 'volume_incidencias_porcentagem', 'porcentagem_certas')
```

O *Serializer* apresentado na Figura 27 realiza os devidos tratamentos nos resultados a serem retornados nas Views. Com os *Serializers* estruturados foram criadas *Views* no arquivo `views.py` que realizam a função de capturar os parâmetros passados nas requisições, os tipos das requisições e realizar as consultas ao banco passando pelos *Serializers*. A Figura 28 apresenta a *View* criada para retornar os resultados a partir de parâmetros de filtros para ano, curso e área de enquadramento.

Figura 28 - View para retorno de resultados com base em filtros.

```
37  class ResultByAnoCursoAndArea(generics.ListAPIView):
38      serializer_class = ResultadoSerializer
39      def get_queryset(self):
40          ano = self.kwargs['ano']
41          curso = self.kwargs['curso']
42          area = self.kwargs['area']
43          return Ft_resultado.objects.filter(ano = ano)
44          .filter(id_curso = curso)
45          .filter(id_area = area)
46          .order_by('qtd_certas', 'qtd_erradas')
```

Com base na *View* apresentada na Figura 28 os dados são ordenados e retornados de acordo com os parâmetros obtidos pela *queryset* e que são usados na filtragem dos resultados.

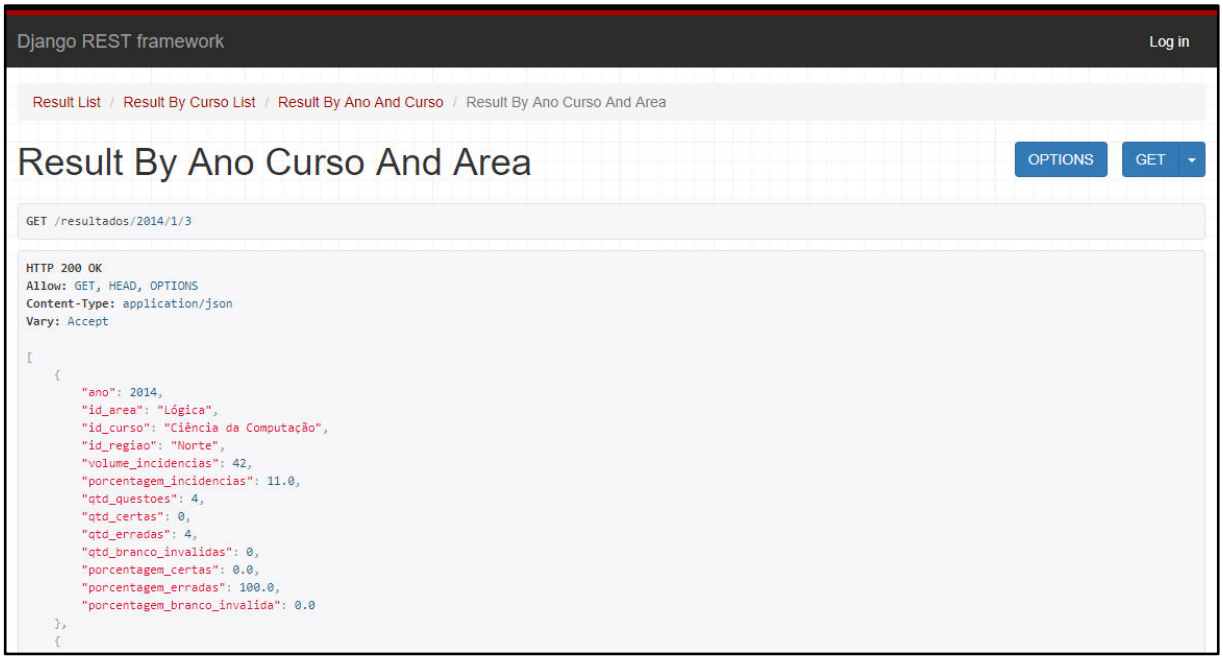
De maneira a obter estes filtros foram configuradas as rotas em uma variável `urlpatterns` em que os filtros são passados como parâmetro diretamente na URL pelas requisições do *Front-end* e encaminhados a suas devidas *Views* para retornarem os dados. A Figura 29 exibe as rotas de requisição criadas no arquivo `urls.py`.

Figura 29 - Rotas para requisições do EnadeDM

```
5   urlpatterns = [
6       path('resultados', views.ResultList.as_view()),
7       path('areas', views.AreaList.as_view()),
8       path('cursos', views.CursosList.as_view()),
9       path('regioes', views.RegioesList.as_view()),
10      path('anos', views.AnoList.as_view()),
11      path('resultados/<int:ano>/<int:curso>/<int:area>', views.ResultByAnoCursoAndArea.as_view()),
12  ]
```

Observando a Figura 29 pode-se notar que os parâmetros para filtragem são do formato inteiro e são passados na rota da linha 11 separados por "/". Assim, um exemplo de url de requisição contendo estes parâmetros seria "resultados/2008/2/3", em que seus parâmetros significam ano, curso e área de enquadramento, respectivamente. Concluídas as configurações da API foram executados testes de retorno para a API localmente. A Figura 30 ilustra o retorno da API para os dados de respostas de estudantes de Ciência da Computação à área de Lógica em 2014.

Figura 30 - Exemplo de retorno da API para os dados de respostas de CC em Lógica em 2014.

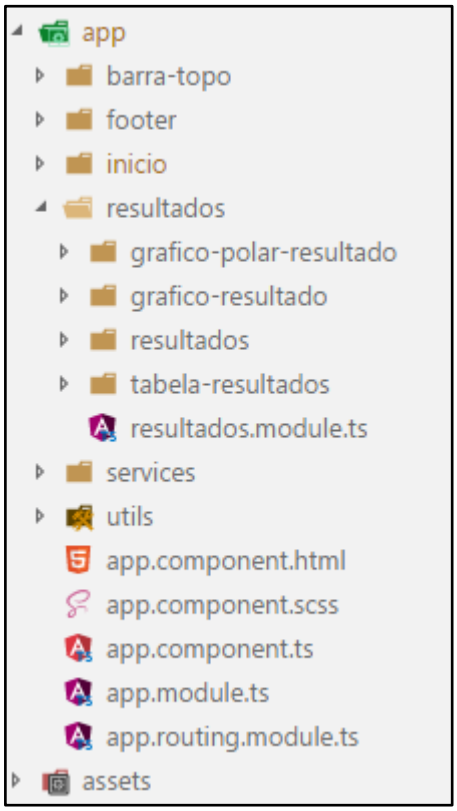


Conforme apresentado na Figura 30, a consulta por método GET realizada à API feita em Django Rest retorna os valores em formato JSON como configurado nos *Serializers*, o que facilita a utilização das respostas e tratamento destas para a sua exibição no *Front-end* por meio de gráficos e tabelas.

4.7.2 Desenvolvimento do Front-end

Para a disponibilização dos dados retornados da API e construir o *Front-end* foi utilizado o Angular 6, que permite a estruturação de um projeto de maneira a dividi-lo em componentes. Diante disso, foram utilizadas bibliotecas do framework que possibilitassem a apresentação destes resultados de forma mais lúdica para facilitar o entendimento que auxiliou na construção de através de gráficos e tabelas. A plataforma foi desenvolvida utilizando dois módulos que incorporam componentes da aplicação, um módulo geral padrão do framework e um módulo `resultados` responsável pela centralização dos componentes de gráficos e tabelas. A Figura 31 apresenta a estrutura de pastas do projeto.

Figura 31 - Estrutura de arquivos do projeto da plataforma.



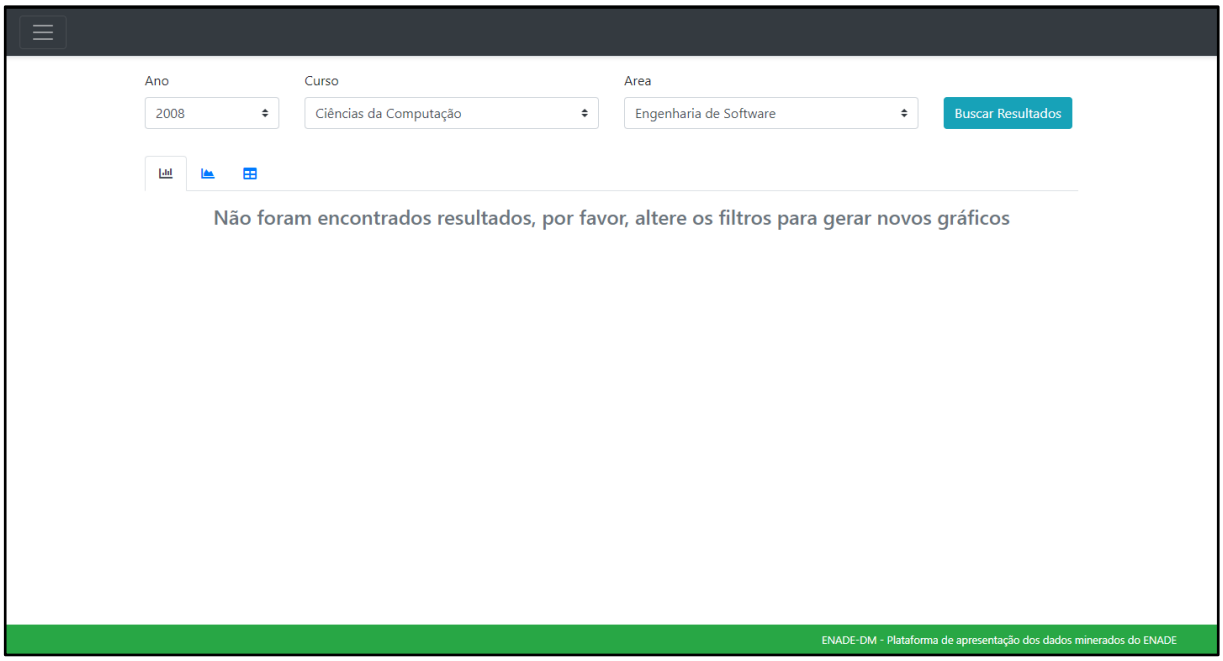
Ao se aplicar a componentização do projeto, foi elaborada a divisão deste em diversos componentes que são estruturados para áreas específicas de apresentação, sendo que neste

trabalho estes foram divididos em: barra do topo, footer, início, resultados, gráfico polar, gráfico de barras e tabela de resultados. A Figura 32 apresenta uma captura de tela da página inicial do EnadeDM.

Figura 32 - Tela Inicial da plataforma.

A Figura 32 apresenta os componentes da tela inicial da plataforma. A parte em cinza contida na figura se trata do componente barra topo (A), que compreende a barra de menus superior da plataforma exibindo os links para navegação entre as páginas da plataforma (início e resultados). Logo abaixo dele há o componente de Início (B), que apresenta um texto curto explicativo sobre a plataforma e sua finalidade, de modo a introduzir o contexto em que a plataforma é aplicada e sua utilização neste contexto. Por fim, tem-se o componente da barra inferior (C) que traz uma frase informativa sobre a plataforma. A segunda tela desenvolvida na plataforma é a tela de resultados, esta tela compreende a apresentação dos campos de filtragem dos dados a serem retornados pela API, bem como gerar os gráficos e tabelas para apresentação dos resultados. A Figura 33 introduz a tela de resultados e seus filtros.

Figura 33 - Tela de resultados.



Ao observar a Figura 33, nota-se que a tela de resultados apresenta três possibilidades de filtros que podem ser utilizados para gerar os gráficos correspondentes às respostas dos estudantes. O primeiro filtro apresentado é o de Ano, em que se é permitido escolher entre um dos três anos analisados neste trabalho. O segundo filtro é o curso a ser selecionado para análise, sendo assim limitados aos três cursos da área de Computação avaliados pelo ENADE para esses anos. Por fim tem-se o filtro por área, que permite a seleção de uma área de enquadramento das questões de modo que seja feita a consulta a API por meio destes três filtros e assim sejam gerados os gráficos para apresentação.

A partir da biblioteca Ngx-Charts foi possível a criação dos gráficos referentes aos dados dos resultados. Cada coluna no gráfico representa uma quantidade de acerto, em que no caso de Lógica para CC em 2008 são quatro questões, em função disso são apresentadas as quantidades de incidências para cada situação. O código apresentado na Figura 34 ilustra a configuração para criação do gráfico de barras de acertos.

Figura 34 - Configuração do componente do Ngx-Charts para criação dos gráficos de barras.

```
<div class="my-5 h-500">
  <ngx-charts-bar-vertical-2d
    [scheme]="colorScheme"
    [results]="dadosResultado"
    [gradient]="gradient"
    [xAxis]="showXAxis"
    [yAxis]="showYAxis"
    [legend]="showLegend"
    [showDataLabel]="true"
    [legendTitle]="legendTitle"
    [roundDomains]="true"
    [showXAxisLabel]="showXAxisLabel"
    [showYAxisLabel]="showYAxisLabel"
    [xAxisLabel]="xAxisLabel"
    [yAxisLabel]="yAxisLabel">
  </ngx-charts-bar-vertical-2d>
</div>
```

Analisando a Figura 34 nota-se que a tag html do componente do NgxCharts está situada na segunda linha e nela são passadas como parâmetros as configurações para criação dos gráficos. Ao definir a paleta de cores do gráfico, foram utilizados tons de verde para gráficos de acertos, tons de vermelho para os gráficos de erros e tons de azul para os gráficos de respostas em branco. Para criar os gráficos a partir dos dados da API foi necessário seguir um formato de dados de acordo com a documentação da biblioteca[6]. A Figura 35 apresenta um exemplo de estrutura a ser seguida.

Figura 35 - Formato dos dados para criação dos gráficos.

```
[
  {
    "name": "Região",
    "series": [
      {
        "name": "Quantidade Certas X",
        "value": 7300000
      },
      {
        "name": "Quantidade Certas Y",
        "value": 8940000
      },
      ...
    ]
  },
  ...
]
```

[6] https://swimlane.gitbook.io/ngx-charts/

Conforme o apresentado na Figura 35, a estrutura compreende de um vetor de objetos compostos de nome e valor para a criação dos grupos gerais do gráfico, em que, para cada objeto deste vetor seu valor é outro vetor de objetos compostos de nome e valor. Para este trabalho os grupos gerais são compostos pelas regiões, em que seus valores são atribuídos a quantidade de respostas acertadas (em número e porcentagem), e o volume de incidências. Este formato é definido pela biblioteca a fim de se agrupar elementos que pertencem a um mesmo elemento pai, como pode ser observado na Figura 36 em que a quantidade de respostas certas é agrupada pelas regiões.

Figura 36 - Gráfico de Volume de incidências em função da quantidade de acertos para as respostas do curso de CC à área de Lógica em 2011.

Conforme apresentado na Figura 36, tem-se a direita do gráfico (C) a legenda que compõe a quantidade de questões, as cores e o número de questões certas para cada coluna gerada. Além do gráfico para os acertos dos acadêmicos, também são gerados gráficos de barras para os erros e respostas em branco, de maneira que se pudesse analisar os dados a fim de verificar novos padrões. Como exemplificado na Figura 36 (B), os gráficos de barra gerados são compostos pelo número de incidências por região em função da quantidade de questões certas marcadas pelos respondentes.

A fim de complementar a apresentação dos resultados, também foi utilizado um gráfico de polarização para apresentação dos resultados. Este pode ser acessado ao se clicar no segundo ícone da barra acima do gráfico (A) apresentado na Figura 36. A criação dos

gráficos utiliza da mesma biblioteca ngx-Charts e para a estruturação destes gráficos o componente recebe os dados como parâmetros de seu componente pai de modo que este incorpora e trata os dados para criação dos gráficos.

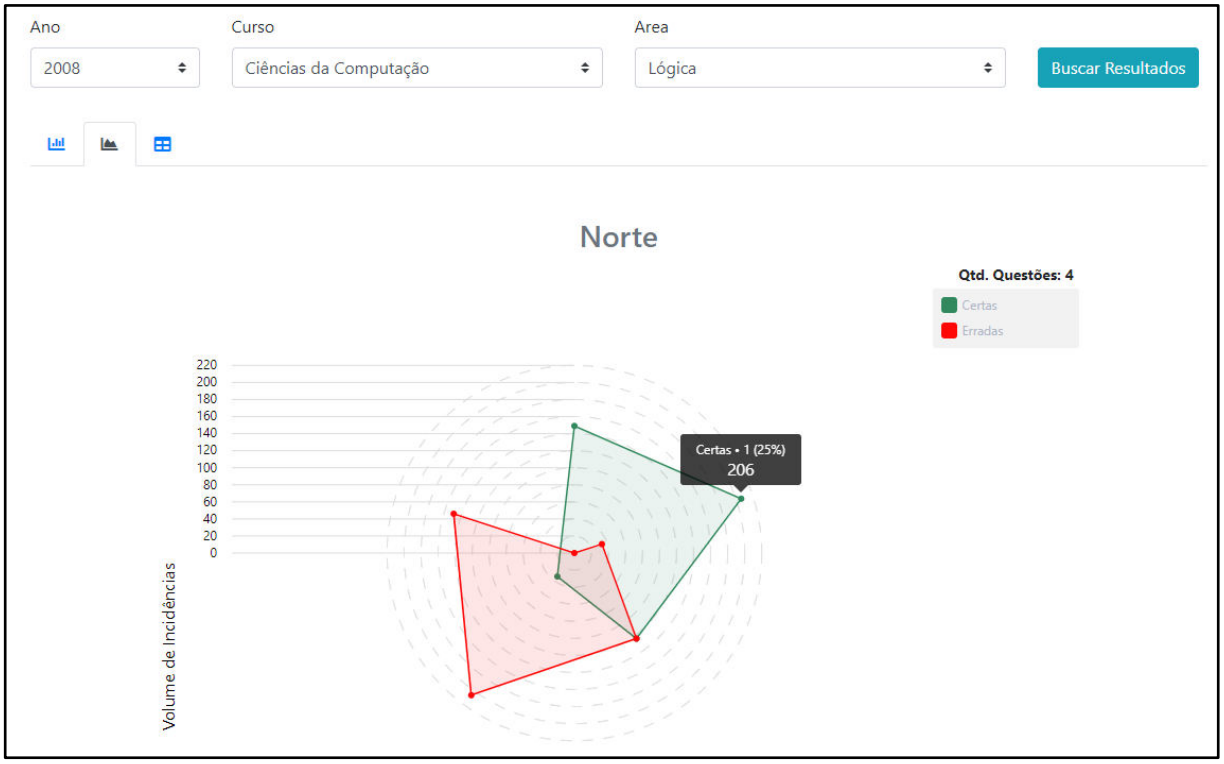
Para a criação deste tipo de gráficos foi utilizada a mesma estrutura de formatação dos dados que a usada para os gráficos de barras apresentada na Figura 36. Contudo, devido ao gráfico apresentar as informações em pontos sobre uma área a estrutura foi ligeiramente alterada, como pode ser observado na Figura 37.

Figura 37 - Estrutura de dados para a criação dos gráficos de polarização.

```
[
  {
    regiao: "Norte",
    dados: [
      {
        name: "Certas",
        series: [
          {
            "name": "Quantidade Certas X",
            "value": 7300000
          },
          ...
        ]
      },
      {
        name: "Erradas",
        series: [
          {
            "name": "Quantidade Erradas X",
            "value": 7300000
          },
          ...
        ]
      },
    ]
  },
  ...
]
```

Diferentemente dos gráficos de barras em que foram gerados dois gráficos distintos para certas, erradas e em branco, para o gráfico de polarização os valores de certas e erradas foram combinados em um único gráfico para cada região, descartando assim o número de brancos por conta de prejudicar a visualização dos gráficos devido ao fato destes estarem mais voltados à comparação dos acertos e erros. Deste modo, foi criado um vetor de regiões que tem por valor dois vetores, sendo que o primeiro armazena as questões certas e o segundo às questões erradas. A Figura 38 apresenta o gráfico de polarização para a região Norte utilizando dos mesmos filtros usados para gerar os gráficos de barras.

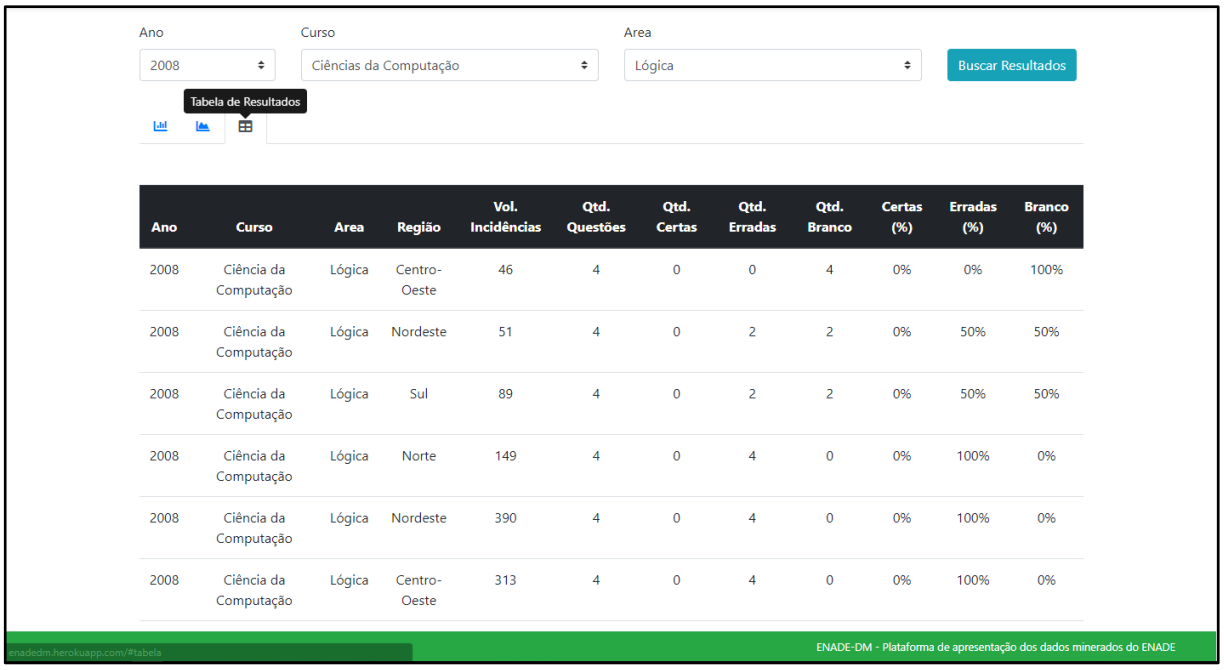
Figura 38 - Gráfico de polarização para região Norte da EnadeDM.



O gráfico de polarização apresenta o contraste entre a quantidade de acertos e erros nas respostas dos dos acadêmicos de Computação. Como apresentado na Figura 38, ao se passar o mouse sobre um ponto do gráfico este apresenta suas características, no caso, se é "certas" ou "erradas", o número de questões e sua porcentagem, e o volume de incidências que se encaixam nessas características. Os gráficos para cada região são dispostos individualmente a fim de facilitar o entendimento da informação neles contidas, pois caso fossem agrupados poderiam dificultar a visualização individual dos resultados.

De maneira a complementar o entendimento dos resultados e, principalmente, dos dados que compõem os gráficos, foi gerado um componente que apresenta a tabela dos resultados obtidos do BD. Deste modo a Figura 39 ilustra a forma de apresentação da tabela de resultados na plataforma gerada através dos mesmos parâmetros de filtros utilizados nos exemplos acima.

Figura 39 - Tabela de resultados na EnadeDM.



Ano	Curso	Area	Região	Vol. Incidências	Qtd. Questões	Qtd. Certas	Qtd. Erradas	Qtd. Branco	Certas (%)	Erradas (%)	Branco (%)
2008	Ciência da Computação	Lógica	Centro-Oeste	46	4	0	0	4	0%	0%	100%
2008	Ciência da Computação	Lógica	Nordeste	51	4	0	2	2	0%	50%	50%
2008	Ciência da Computação	Lógica	Sul	89	4	0	2	2	0%	50%	50%
2008	Ciência da Computação	Lógica	Norte	149	4	0	4	0	0%	100%	0%
2008	Ciência da Computação	Lógica	Nordeste	390	4	0	4	0	0%	100%	0%
2008	Ciência da Computação	Lógica	Centro-Oeste	313	4	0	4	0	0%	100%	0%

Esta tabela pode ser acessada a partir do terceiro item do menu disposto no menu superior aos gráficos. A tabela apresentada na Figura 39 carrega as informações assim como são dispostas na tabela para inserção dos dados como a apresentada na tabela de resultados da Figura 23. Deste modo, é possível ter uma visão geral dos dados resultantes da mineração dos dados do ENADE.

4.8 DISPONIBILIZAÇÃO DA PLATAFORMA

Como o foco do trabalho está na disponibilização da plataforma para acesso público, não foram utilizadas autenticações ou técnicas relacionadas que pudessem limitar a experiência do usuário ao utilizar a plataforma. Isto se aplica ao contexto geral da plataforma, com exceção ao acesso ao BD pelo Django Admin, que necessita da autenticação de um usuário com permissões de administrador. O Heroku foi utilizado como host da plataforma devido à possibilidade de hospedagem da *Front-end* da plataforma e da API sem custos adicionais.

Para a hospedagem do *Front-end* foi necessário a utilização da versão de produção do projeto em Angular obtida a partir do comando `ng build --prod`. Este comando comprime os arquivos de estilização e configuração do projeto em arquivos javascripts para sua utilização em servidores web. Além da compressão, foi criado um arquivo `index.php` que tem por função redirecionar o acesso da url do servidor para o arquivo .html principal do

projeto. A partir deste arquivo .php o heroku identifica o projeto como sendo PHP e configura o servidor de host como um servidor de conexão Apache.

Para o *deploy* da API no Heroku foram feitas as devidas configurações no arquivo `settings.py` do projeto de modo que os modelos de dados fossem criados no Heroku ao subir o projeto. De modo a assegurar que as dependências do projeto fossem instaladas corretamente no servidor destino foi criado um arquivo `requirements.txt` apresentado na Figura 40 que contém os nomes das dependências e as versões necessárias para seu funcionamento devido.

Figura 40 - Requirements.txt

```
certifi==2018.10.15
chardet==3.0.4
Django==2.1.2
django-cors-headers==2.4.0
djangorestframework==3.9.0
gunicorn==19.9.0
heroku==0.1.4
idna==2.7
python-dateutil==1.5
pytz==2018.5
requests==2.20.0
urllib3==1.24
```

De forma a assegurar que todas as dependências estivessem armazenadas no arquivo foi utilizado o comando `pip freeze` diretamente no terminal. Este comando é recomendado pela documentação do Django Rest Framework para adicionar as dependências necessárias para o funcionamento total do projeto no arquivo de configuração `requirements.txt` do projeto. Finalizada esta configuração foi feito o deploy do projeto da API seguindo a documentação disposta pelo Heroku para projetos em Python.

Um dos problemas enfrentados ao se configurar a integração entre o *Front-end* e a API no Heroku foi o problema de CORS, que não permitia o *Front-end* realizar requisições à API. Para contornar este problema foi feita a instalação e configuração da dependência `django-cors-headers` no projeto da API de modo que este liberasse as requisições para o *Front-end*. A configuração foi feita ao atribuir o valor True (Verdadeiro) à variável da

`"CORS_ORIGIN_ALLOW_ALL"`, deste modo são liberadas as requisições de hosts externos à API.

Com os problemas da API identificados e corrigidos foi feito o deploy do *Front-end* para o servidor no Heroku. A partir do momento em que um projeto do Heroku é criado e possui um deploy feito é gerada uma URL de acesso a este seguindo o padrão: nome_do_projeto.herokuapp.com. Desta forma, a plataforma completa do EnadeDM pode ser acessada pela seguinte URL: http://enadedm.herokuapp.com.

5 CONSIDERAÇÕES FINAIS

O presente trabalho descreveu o processo de mineração, análise e disponibilização dos resultados da análise dos microdados do ENADE das respostas dos estudantes da área da Computação. O desenvolvimento do trabalho teve como objetivo avaliar o desempenho desses estudantes com base nos campos de estudo da Computação nos diferentes cursos de graduação existentes no Brasil e avaliados pelo INEP. Diante disso foi apresentado um referencial teórico sobre o tema abordado, assim como conceitos e características do modelo de referência CRISP-DM, que foi escolhido para apoiar no processo de alcance dos objetivos propostos.

A partir do endereço https://enadedm.herokuapp.com/ é possível visualizar os resultados do presente trabalho, sendo que a plataforma criada apresenta de forma gráfica os resultados obtidos da aplicação dos algoritmos e informações que podem auxiliar na análise das informações obtidas. A aplicação foi disponibilizada publicamente na internet de modo a ser acessada facilmente por qualquer pessoa interessada nos resultados obtidos. A princípio esta será utilizada como forma de visualização das informações, entretanto, seus resultados podem ser utilizados em conjunto com outras pesquisas com foco em regiões ou cursos específicos, de maneira a ampliar a descoberta de informações ou auxiliar em novas pesquisas.

Com as informações obtidas é fornecido aos estudantes da área de Computação um conjunto de dados relevantes, coerentes e úteis para implementação de melhorias nos cursos de graduação da área de Computação das IES brasileiras. A utilização destas informações poderá beneficiar o entendimento de como o foco das disciplinas dos cursos da área de Computação estão sendo passados aos alunos.

Como trabalhos futuros, pretende-se ampliar o escopo da mineração de dados para outras áreas de aplicação do ENADE, como também ampliar a análise nos resultados da mineração ao utilizar diferentes algoritmos e técnicas de mineração dos dados. Por fim, espera-se utilizar aprendizado de máquina de modo a aprimorar o reconhecimento de padrões e informações sobre estes dados.

REFERÊNCIAS

AGRAWAL, Rakesh; SRIKANT, Ramakrishnan. Fast Algorithms for Mining Association Rules. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 94., 1994, São Francisco. **Proceedings Of The 20th International Conference On Very Large Data Bases.** São Francisco: Vldb, 1994. v. 94, p. 487 - 499. Disponível em: <http://www.vldb.org/conf/1994/P487.PDF>. Acesso em: 12 jun. 2018.

AGRAWAL, Rakesh; IMIELINSKI, Tomasz; SWAMI, Arun. Mining association rules between sets of items in large databases. **Proceedings Of The 1993 Acm Sigmod International Conference On Management Of Data**, Washington, v. 20, n. 2, p.207-216, jul. 1993. Anual. Disponível em: <https://dl.acm.org/citation.cfm?id=170072>. Acesso em: 12 maio 2018.

BARTOLOMEU, Tereza Angélica. **Modelo de investigação de acidentes do trabalho baseado na aplicação de tecnologias de extração de conhecimento.** 2002. 302 f. Tese (Doutorado) - Curso de Engenharia, Universidade Federal de Santa Catarina, Florianópolis, 2002. Disponível em: <https://repositorio.ufsc.br/xmlui/bitstream/handle/123456789/83836/189111.pdf?sequence=1&isAllowed=y>. Acesso em: 25 mar. 2018.

BIJURAJ, L.v. Clustering and its Applications. In: NATIONAL CONFERENCE ON NEW HORIZONS IN IT. 2013, [s/l]. **Proceedings Of National Conference On New Horizons.** [s/l]: Bhujbal Knowledge Center, 2013. p. 169 - 172. Disponível em: <http://www.met.edu/Institutes/ICS/NCNHIT/papers/39.pdf>. Acesso em: 25 maio 2018.

CAMILO, Cássio Oliveira; SILVA, João Carlos. **Mineração de Dados**: Conceitos, tarefas, métodos e ferramentas. 2009, [s/l]. Disponível em: <http://www.portal.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-09.pdf>. Acesso em: 15 mar. 2018.

CHAPMAN, Pete et al. **CRISP-DM 1.0:** Step-by-step data mining guide. [s/l]: Spss, 2000. 76 p. Disponível em: <https://www.the-modeling-agency.com/crisp-dm.pdf>. Acesso em: 06 maio 2018.

CARVALHO, Ricardo Silva. **Modelos preditivos para avaliação de risco de corrupção de servidores públicos federais.**2016. 119 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2016. Disponível em: <http://repositorio.unb.br/handle/10482/19361>. Acesso em: 06 jun. 2018.

CAVALCANTI, Gabriela Góis; FELL, André Felipe de Albuquerque; DORNELAS, Jairo Simião. Data Warehouse: uma ferramenta de tecnologia de informação para as organizações.. In: SIMPEP, 12., 2005, São Paulo. **ANAIS DO XII SIMPEP.** São Paulo: Unesp, 2005. p. 1 - 12. Disponível em: <http://www.simpep.feb.unesp.br/anais/anais_12/copiar.php?arquivo=GOIS_GC_Data%20Warehouse.pdf>. Acesso em: 23 out. 2018.

CIELO, Ivã (Ed.). **Data Warehouse como diferencial competitivo.** 2005. Disponível em: <http://www.always.com.br/site2005/internet_clip07.html>. Acesso em: 16 out. 2018.

CÔRTES, Sérgio da Costa; PORCARO, Rosa Maria; LIFSCHITZ, Sérgio. **Mineração de Dados** – Funcionalidades, Técnicas e Abordagens. PUC, 2002. Disponível em: <ftp://obaluae.inf.puc-rio.br/pub/docs/techreports/02_10_cortes.pdf>. Acesso em: 18 abr. 2018.

SILVA, Leandro Augusto da; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à Mineração de dados com aplicações em R.** Rio de Janeiro: Elsevier, 2016. ISBN: 978-85-352-8447-8.

GONSALVES, Eduardo Corrêa. Regras de Associações e suas Medidas de Interesse Objetivas e Subjetivas. **INFOCOMP**, [s/l], v. 4, n. 1, p. 26-35, mar. 2004. ISSN 1982-3363. Disponível em: <http://infocomp.dcc.ufla.br/index.php/INFOCOMP/article/view/79>. Acesso em: 12 mai. 2018.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data Mining:** Concepts and Techniques.. 3. ed. Waltham: Elsevier, 2012. 740 p. Disponível em: <http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf>. Acesso em: 12 maio 2018.

HIRAGI, Gilberto de Oliveira. **Mineração de dados em base de germoplasma.** 2008. 107 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade de Brasília, Brasília, 2008. Disponível em: <http://repositorio.unb.br/handle/10482/1187>. Acesso em: 26 abr. 2018.

IBM. **Visão geral da ajuda do CRISP-DM**. *IBM Knowledge Center.* 2017. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS3RA7_17.1.0/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html.>. Acesso em: 06 mar. 2018

INEP. **Enade**. 2018. Disponível em: <http://portal.inep.gov.br/enade>. Acesso em: 17 abr. 2018.

INMON, William H. **Building the Data Warehouse.** 3. ed. New York: John Wiley & Sons, Inc., 2002. 428 p.

KANTARDZIC, Mehmed. **DATA MINING:** Concepts, Models, Methods, and Algorithms. 2th. ed. New Jersey: Ieee Press, 2011. 550 p. Disponível em: <https://doc.lagout.org/Others/Data Mining/Data Mining_ Concepts, Models, Methods, and Algorithms (2nd ed.) [Kantardzic 2011-08-16].pdf>. Acesso em: 13 abr. 2018.

LIU, Bing. **Web Data Mining:** Exploring Hyperlinks, Contents, and Usage Data. 2. ed. Chicago: Springer, 2007. 643 p. Disponível em: <http://sirius.cs.put.poznan.pl/~inf89721/Seminarium/Web_Data_Mining__2nd_Edition__Exploring_Hyperlinks__Contents__and_Usage_Data.pdf>. Acesso em: 12 maio 2018.

MORO, Sérgio; LAUREANO, Raul; CORTEZ, Paulo. **Using data mining for bank direct marketing: an application of the CRISP-DM methodology.** In: EUROPEAN SIMULATION AND MODELLING CONFERENCE, 25., 2011, Guimarães. Proceedings of European Simulation and Modelling Conference - ESM'2011. Guimarães: Esm, 2011. p. 117 - 121. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/14838>. Acesso em: 23 mar. 2018.

OLSON, David L; DELEN, Dursun. **Advanced Data Mining Techniques.** Berlin: Springer, 2008. 182 p. Disponível em: <https://pdfs.semanticscholar.org/c1c7/4829d6430d468a1fe1f75eae217325253baf.pdf>. Acesso em: 13 abr. 2018.

PELEGRIN, Diana Colombo et al. **A Shell de Data Mining Orion: Classificação, Clusterização e Associação**. 2012. Anais SULCOMP, v. 1. Disponível em: <http://periodicos.unesc.net/sulcomp/article/view/799>. Acesso em: 18 abr. 2018.

ROMERO, Cristobal; VENTURA, Sebastian. **Educational data mining: A survey from 1995 to 2005**. Expert Systems With Applications. Cordoba, p. 135-146. [s/l]. 2007. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417406001266?via=ihub>. Acesso em: 13 maio 2018.

SOUZA FILHO, Hélcio Gomes de. **EXTRAÇÃO DE REGRAS DE ASSOCIAÇÃO EM UMA BASE DE DADOS RELACIONAL.** 2004. 74 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004. Disponível em: <http://www.coc.ufrj.br/es/component/docman/?task=doc_download&gid=1789&Itemid=>.>. Acesso em: 05 jun. 2018.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao DATAMINING Mineração de Dados**. Ciência Mo ed. Rio de Janeiro: Ciência Moderna Ltda, 2009. 296 p.

VASCONCELOS, Lívia Maria Rocha de; CARVALHO, Cedric Luiz de. **Aplicação de Regras de Associação para Mineração de Dados na Web**. Universidade Federal de Goiás, Goiânia, 2004. 20 p. Disponível em: <http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_004-04.pdf>. Acesso em: 12 maio 2018.

VOLZNIAK, Fabricio; VIANA, Leonardo. **Data Mining Classification.** CSEP 521 - University of Washington. 2007. Disponível em: <https://courses.cs.washington.edu/courses/csep521/07wi/prj/leonardo_fabricio.pdf >. Acesso em: 15 mar. 2018