Universidade Federal do Rio de Janeiro - UFRJ
Nome: Ana Clara Monteiro de Oliveira
DRE: 115199661
MAB602: Data Warehousing no Suporte à Tomada de Decisão - 2021/1
Professor: Geraldo Xexéo
Relatório Individual

O presente relatório visa apresentar uma análise dos dados relacionados ao Exame Nacional de Desempenho dos Estudantes (Enade) dos anos de 2017, 2018 e 2019. Para a análise de dados foi utilizada a ferramenta *KNIME*, onde foi feita a leitura, tratamento de dados e geração de gráficos.

Questão 1

Um fluxo pelo *KNIME* foi utilizado para baixar os arquivos e fazer a leitura dos mesmos. Para a leitura foi utilizado o nó "*Unzip Files (legacy)*" e para a leitura dos arquivos foi utilizado o nó "*CSV Reader*".

Os arquivos foram baixados do site oficial do INEP - ENADE[1]. Durante a realização do *download* automático, percebeu-se que o link para 2019 estava quebrado e não foi possível realizar essa etapa. Então, para facilitar a leitura, foi decidido baixar os arquivos manualmente, descompactá-los, separar somente os arquivos com os dados, compactá-los, e então subir para o Github[2].

Ao baixá-los, cada ano vinha com 3 pastas: "DADOS", "INPUTS" e "LEIA-ME". A pasta de "DADOS" contém um arquivo .txt com todas as informações do ENADE relacionados àquele ano. A pasta "INPUTS" contém códigos na linguagem R, *sas* e *sps* para abrir esses dados nesses programas. Já a pasta "LEIA-ME" contém o dicionário de dados, manual do usuário (PDF) e o questionário do estudante (PDF).

Para o download e leitura dos arquivos, foram feitos os seguintes passos:

1- Criando um *workflow* para o fluxo do trabalho: especificando seu nome e onde ele salvará o fluxo *KNIME*. Optou-se por salvar o workflow no destino local do *KNIME*, fazendo com que qualquer pessoa que tenha acesso a esse fluxo não tenha problemas em encontrar estes arquivos.

[1] https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/enade
[2] https://github.com/acmont/Prova-individual---Data-Warehouse-21.1

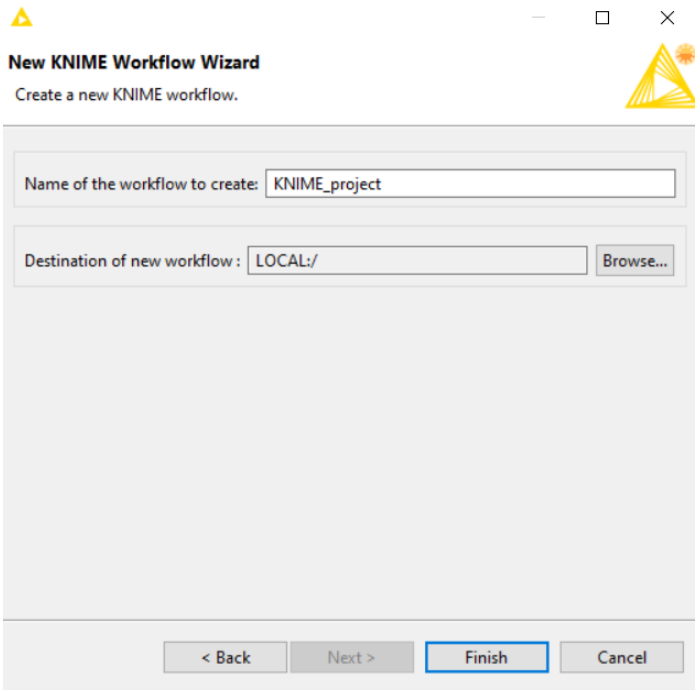

Figura 1- Criação do workflow

2- Descompactando os arquivos: selecionando o link do github, onde o arquivo está compactado, e salvando o arquivo descompactado na pasta local com o nome do workflow.

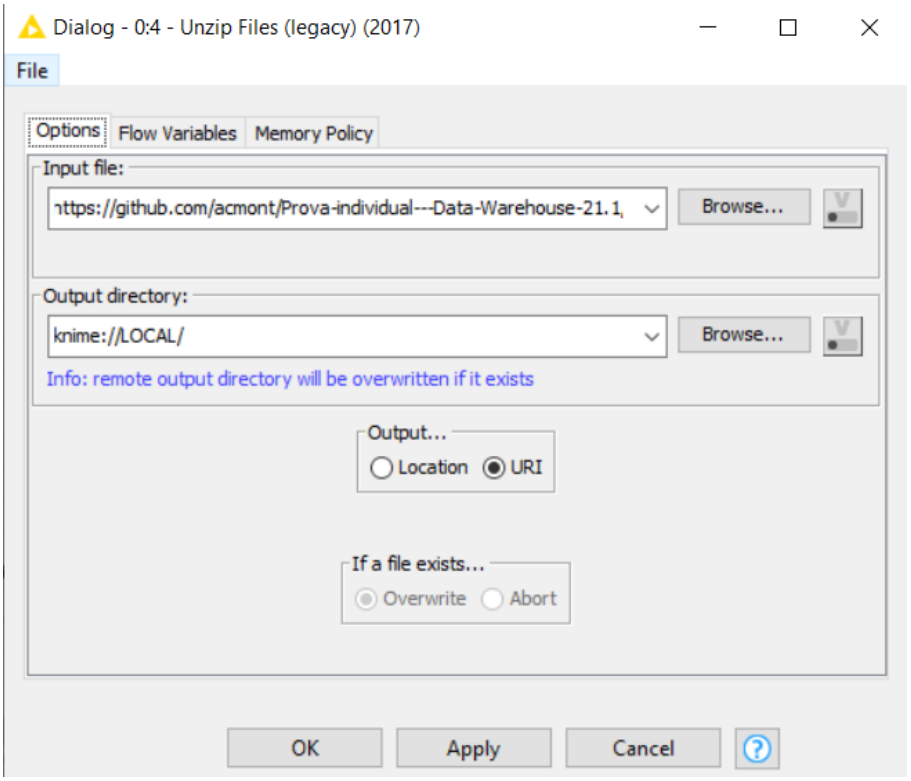

Figura 2 - Descompactação dos arquivos do ano de 2017.

3- Leitura dos dados: Para conseguir fazer a leitura dos arquivos descompactados na pasta correta, o nó foi configurado como mostra a figura 3. Então, foi ajustado o formato do arquivo, para que pudesse ser feita a análise do mesmo.

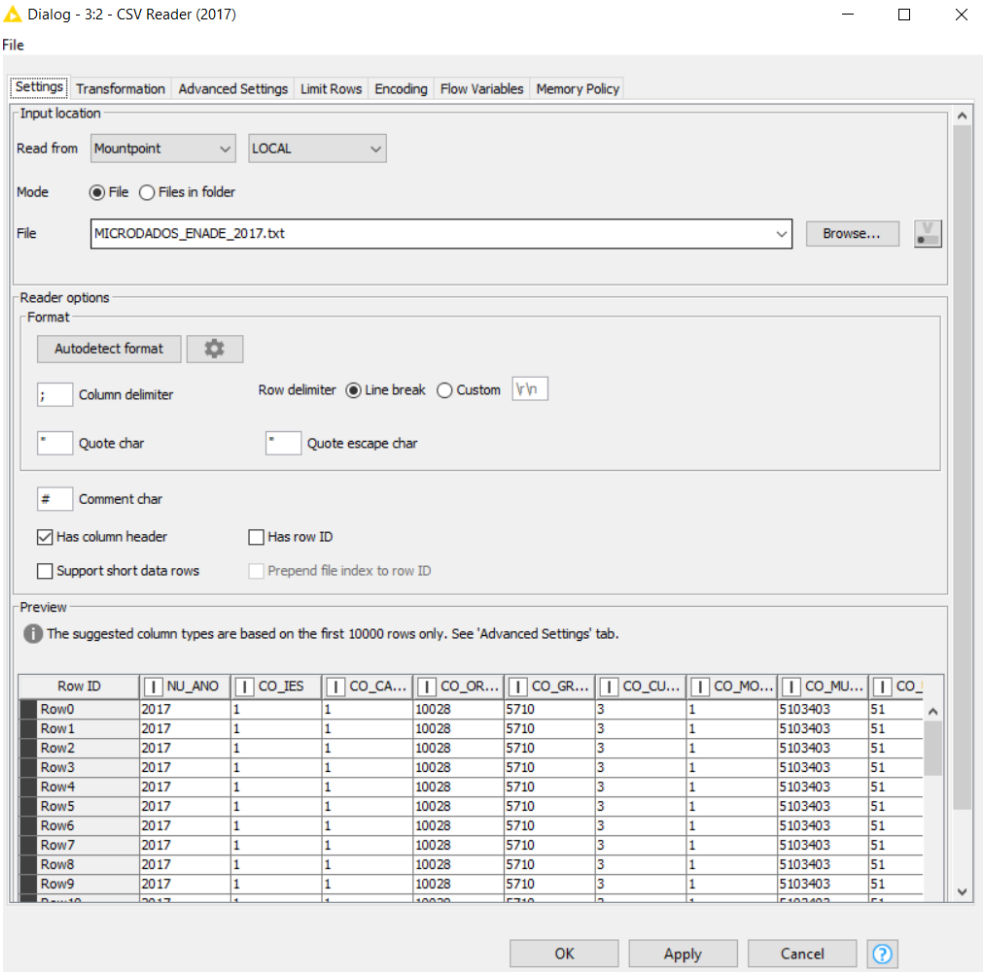

Figura 3 - Leitura dos dados

Após a descompactação e leitura dos dados, foi adicionado um nó para concatenar os 3 arquivos do ENADE (anos de 2017, 2018 e 2019), como mostra a figura abaixo:

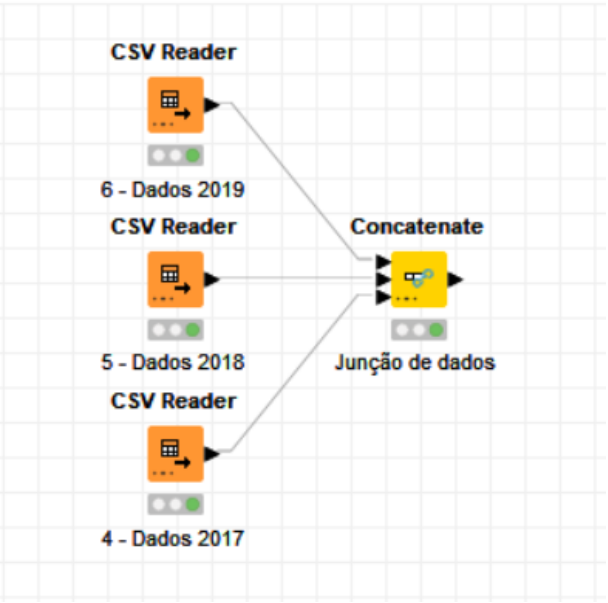

Figura 4 - Leitura dos dados

Questão 2

Para montar o modelo dimensional estrela, foi consultado um dicionário de dados que está disponível junto com os microdados de cada ano. O dicionário é o mesmo para os anos de 2017, 2018 e 2019. O modelo dimensional refere-se aos três anos, uma vez que estes possuem o mesmo dicionário de dados. O modelo estrela consiste em uma tabela fato e dez dimensões.Como o dicionário de dados do ENADE veio dividido em categorias, a ramificação para criar as dimensões ficou um pouco mais simples. Para uma melhor visualização da imagem, a mesma está no Github[3].

Para a criação do modelo estrela foi utilizada a ferramenta *Db Diagram* (https://dbdiagram.io/d), onde o *script* se encontra no anexo [1].

[3] https://github.com/acmont/Prova-individual---Data-Warehouse-21.1/tree/main/Gráficos%20e%20tabelas

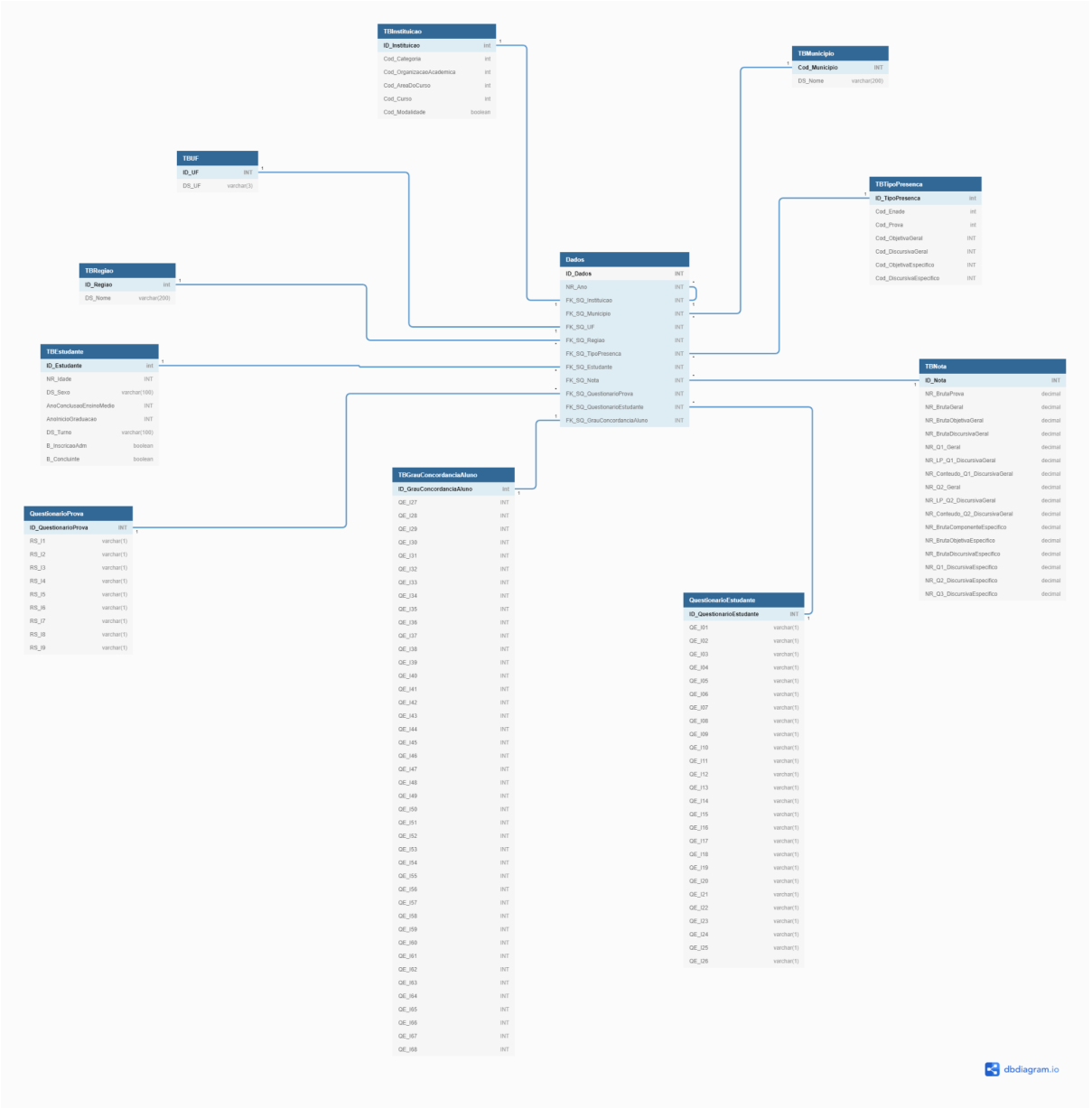

Figura 5 - Modelo Estrela ENADE

Questão 3

Para a criação da base de dados no MySQL (versão 5.7, pois é umas das versões que o KNIME utiliza), foi utilizado um servidor externo para fosse possível trabalhar com a grande carga de dados, por conta da capacidade de trabalho da máquina. O nome do banco de dados criado é “dados_enade”, onde foram criadas as tabelas: “QuestionarioEstudante”, “QuestionarioProva”, “TBEstudante”, “TBGrauConcordanciaAluno”, “TBInstituicao”, “TBMunicipio”, “TBNota”, “TBRegiao”, “TBTipoPresenca”, “TBUF”.

A ferramenta utilizada para a criação do banco de dados foi o *DataGrip*, e então foi gerado um *script* de acordo com o modelo construído acima. O *script* da estrutura do banco se encontra na pasta arquivos do Github [2].

Questão 4

Assim como no trabalho em grupo, foi feito um fluxo no *KNIME*, onde com apenas um comando o usuário faria o carregamento dos dados.

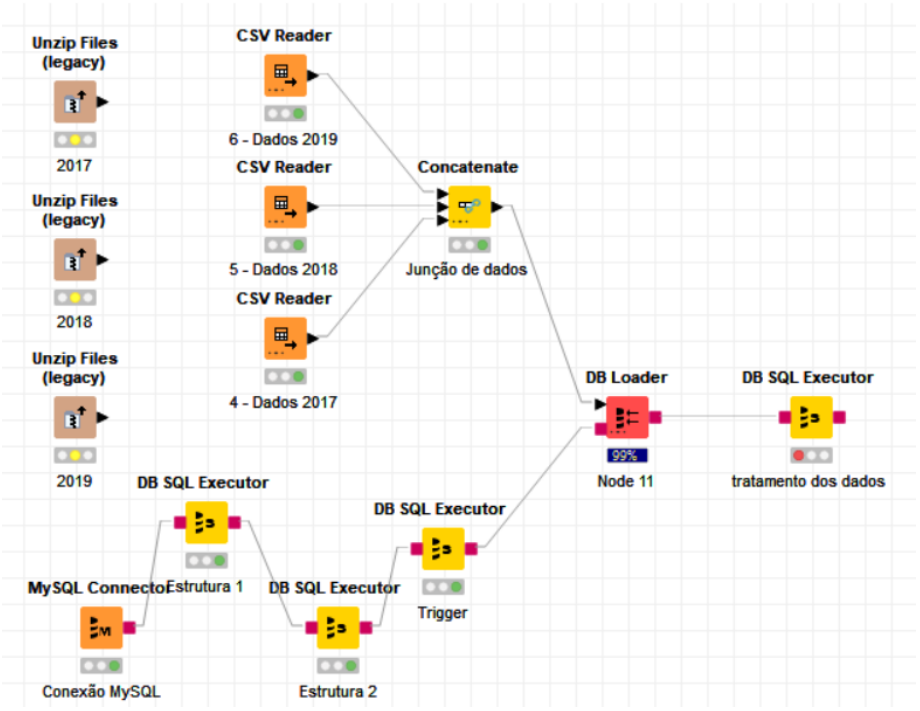

Figura 6 - Fluxo para a carga dos dados.

Para o carregamento de dados foi criada uma tabela auxiliar, "enade_aux", assim facilitando o *input* dos dados. A tabela *enade_aux* possui os mesmos campos e a mesma ordem que os arquivos originais do ENADE baixados da página do INEP, de acordo com o dicionário de dados. Para facilitar a inserção, todos os tipos das variáveis entraram como 'varchar()'.

Após a inserção dos dados na tabela auxiliar, foi feito um tratamento para que estes estivessem conforme o modelo relacional desenvolvido, através de um script em SQL que se encontra no repositório do Github (individual_1.sql e individual_2.sql)[4]. Para esse tratamento, foi feito um *in*sert para colocar os questionários tanto do estudante quanto da prova, notas dos alunos e grau de concordância. Alguns campos como, 'Região', 'UF', e 'Município' foram inseridas manualmente com seus respectivos códigos de identificação de acordo com o dicionário disponibilizado.

Abaixo serão descritos os nós utilizados no fluxo:

- MySQL Connector (Conexão MySQL): Este nó conecta com o banco local criado para armazenar os dados carregados.
- DB SQL Executor (Estrutura 1): Este nó contém o código em SQL que cria o database 'ana_dwIndividual';

[4] https://github.com/acmont/Prova-individual---Data-Warehouse-21.1/tree/main/scripts%20banco - Script do carregamento de dados.

- DB SQL Executor (Estrutura 2): Este nó contém o código em SQL que cria a tabela auxiliar contendo todos os dados como varchar();

- DB SQL Executor (Trigger): Este nó contém o código em SQL que cria uma *trigger* que otimiza a inserção dos dados na tabela auxiliar.

- DB Loader (load data from csv): Este nó faz a carga dos dados para o banco de dados local, onde é configurado tanto o banco de dados quanto a tabela auxiliar (enade_aux);

- DB SQL Executor (início do tratamento de dados): Inicialmente, os dados vazios (‘’, na, ‘ ’) foram tratados como valores nulos. Em seguida foram acrescentadas as instituições identificadas no banco de dados, tipos de presença e os dados do estudante no banco relacional. Após todo o tratamento, a tabela auxiliar foi deletada restando apenas as tabelas do modelo estrela no banco.

Questão 5

A visualização dos dados foi feita pela ferramenta *Tableau desktop*. O banco de dados foi conectado ao Tableau e assim foi possível gerar os gráficos e tabelas listados abaixo. As imagens e tabelas em CSV se encontram no repertório do Github. As perguntas para as análises foram as seguintes:

a. Qual cor ou raça é predominante no Rio de Janeiro para os inscritos nos anos de 2017, 2018 e 2019?

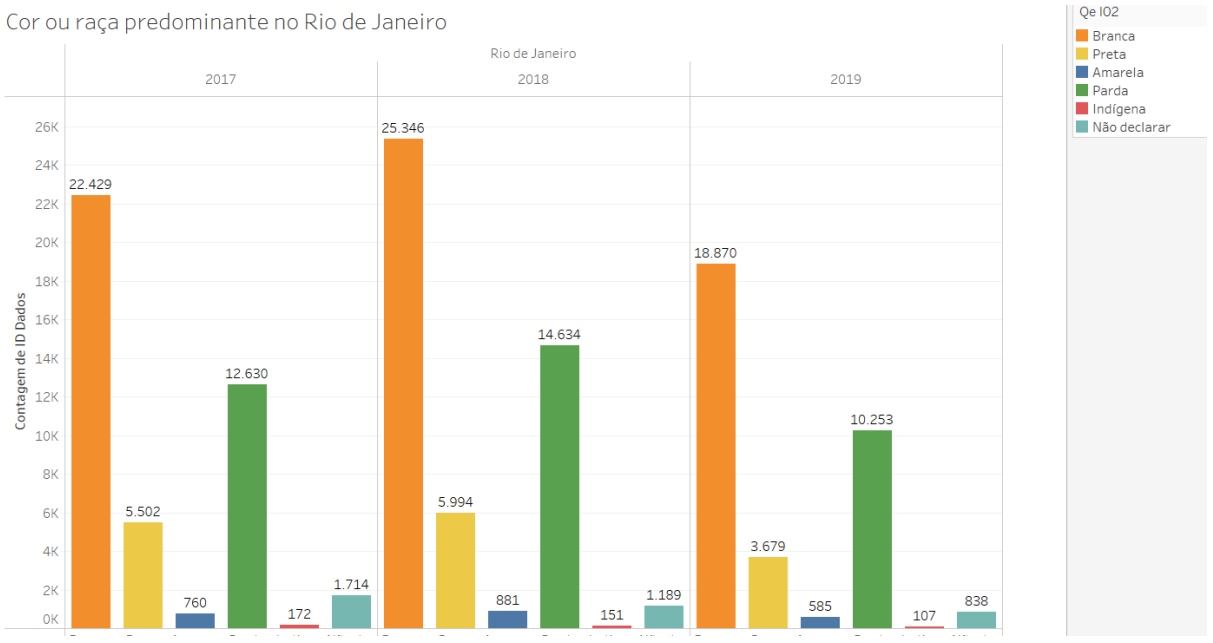

Figura 7 - Gráfico de cor ou raça predominante no Rio de Janeiro

Pela análise do gráfico acima (figura 5), a cor Branca é predominante pelos inscritos por todos os anos, seguido pela cor parda.

b. Qual foi a média das notas de todos os participantes por ano?

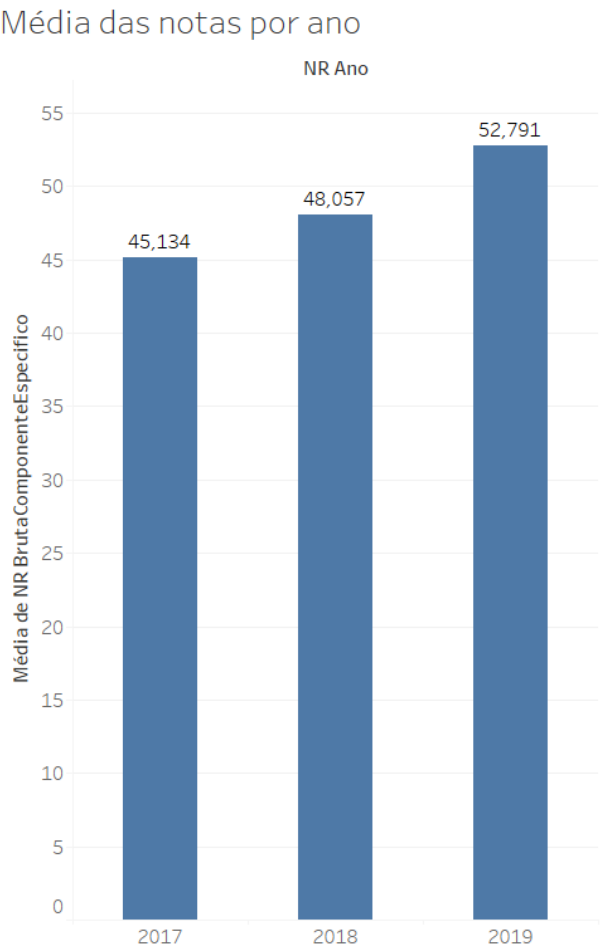

Figura 8 - Gráfico das médias por ano.

Percebe-se que ao longo dos anos foi possível ter um crescimento na nota dos participantes, mostrando um melhor desempenho.

c. Qual o maior grau de dificuldade ao longo dos anos?

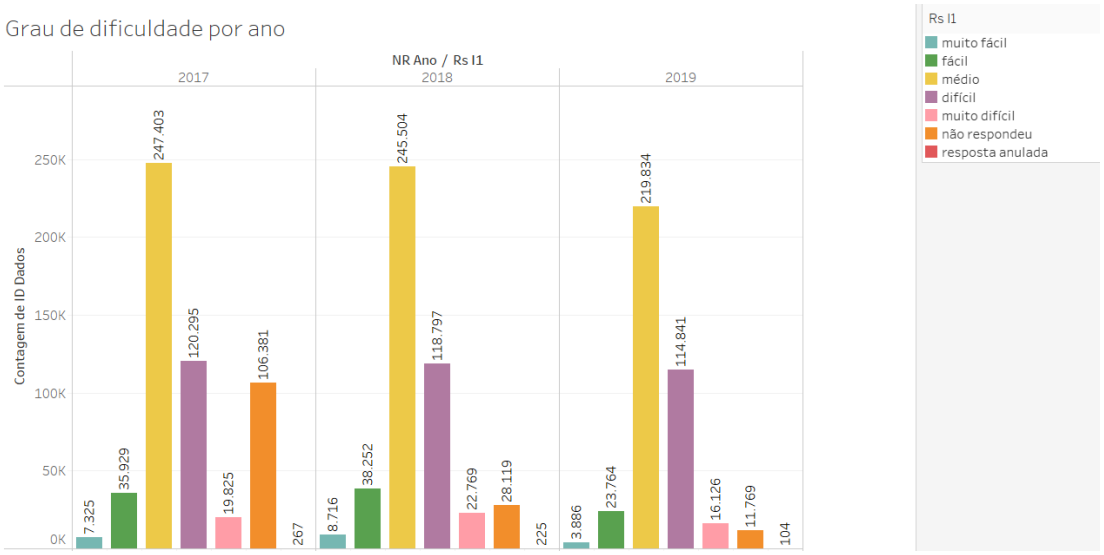

Figura 9 - Gráfico do maior grau de dificuldade

O grau de dificuldade predominante nos 3 anos analisados, é o grau de dificuldade médio.

d. Qual a média das notas de acordo com a situação de trabalho dos participantes?

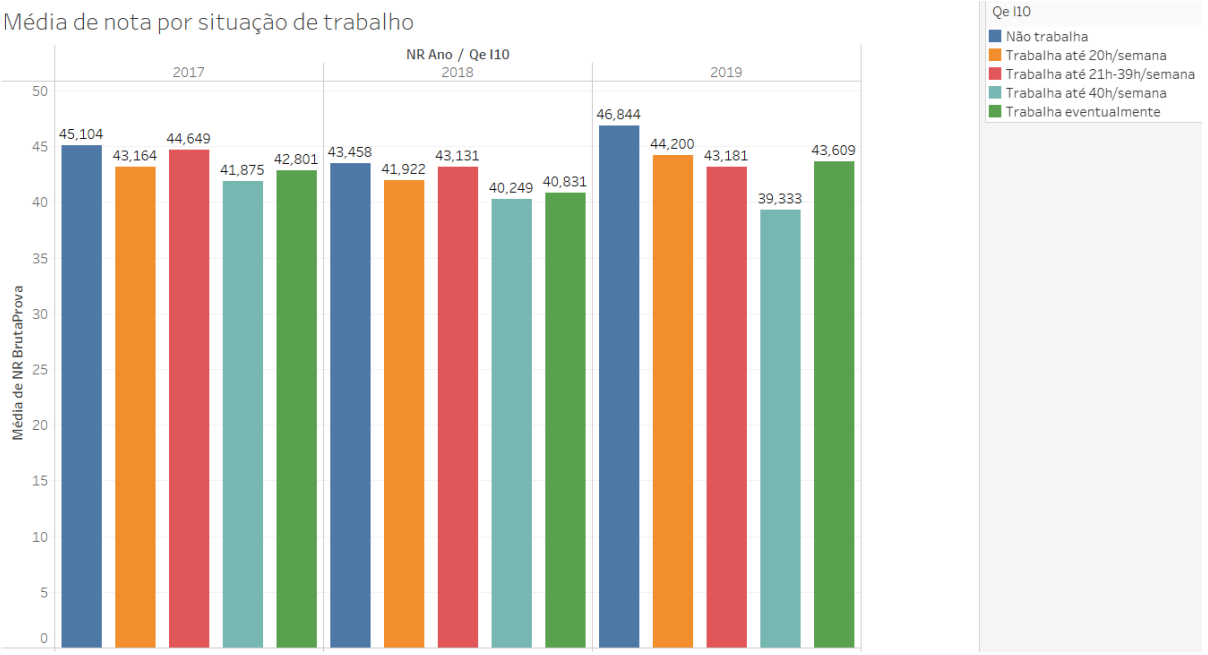

Figura 10 - Gráfico das médias por situação de trabalho.

Apesar das médias estarem bem próximas, os estudantes que não possuem a situação de trabalho como “Não trabalha” são os que possuem médias maiores em todos anos.

e. Qual a região com as melhores médias de notas em 2017, 2018 e 2019?

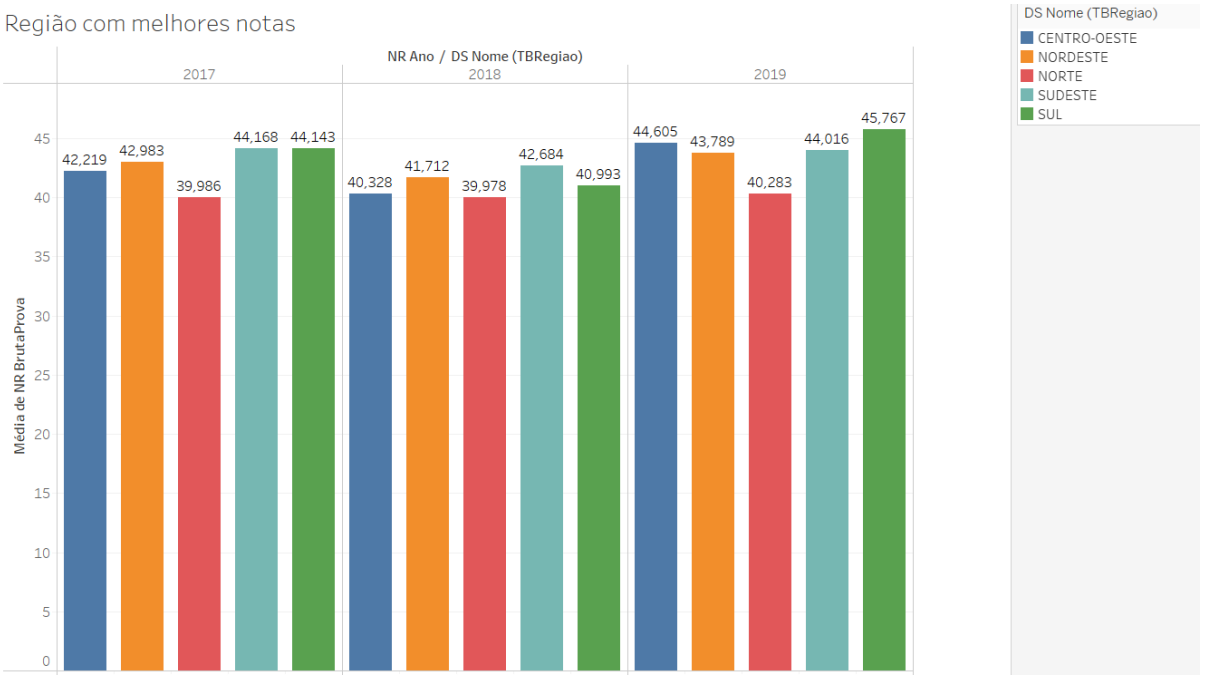

Figura 11 - Gráfico das médias das notas por Região

Nos dois primeiros anos, 2017 e 2018, a região Sudeste teve as melhores notas. No ano de 2019 a região Sul obteve melhor desempenho em relação às notas.

Questão 7

Abaixo serão listadas as ferramentas utilizadas para a realização deste trabalho.

- Db Diagram: https://dbdiagram.io/d - Essa ferramenta foi escolhida por conta da sua facilidade em montar o modelo dimensional. A mesma foi utilizada para o trabalho em grupo também.
- DataGrip: https://www.jetbrains.com/datagrip/ - Essa ferramenta foi escolhida por ter sido utilizada no trabalho em grupo, pela sua maior facilidade e praticidade em relação ao MySQL *workbench*.
- Knime: https://www.knime.com/ - Essa ferramenta foi utilizada para fazer o download automático dos dados.
- Tableau: https://www.tableau.com/pt-br - Essa ferramenta é muito utilizada para a geração de gráficos e tabelas. A versão utilizada foi o *desktop*.
- GitHub: **https://github.com/acmont/Prova-individual---Data-Warehouse-21.1** - O GitHub foi utilizado para colocar o repositório dos dados analisados.

ANEXOS

[1] Script modelo estrela - ENADE

```
Table TBInstituicao{
  ID_Instituicao int [pk, increment, unique] // auto-increment
  Cod_Categoria int
  Cod_OrganizacaoAcademica int
  Cod_AreaDoCurso int
  Cod_Curso int
  Cod_Modalidade boolean
}


Table TBEstudante{
  ID_Estudante int [pk, increment, unique] // auto-increment
  NR_Idade INT
  DS_Sexo varchar(100)
  AnoConclusaoEnsinoMedio INT
  AnoInicioGraduacao INT
  DS_Turno varchar(100)
  B_InscricaoAdm boolean
  B_Concluinte boolean
}

Table TBTipoPresenca{
  ID_TipoPresenca int [pk, increment, unique] // auto-increment
  Cod_Enade int
  Cod_Prova int
  Cod_ObjetivaGeral INT
  Cod_DiscursivaGeral INT
  Cod_ObjetivaEspecifico INT
  Cod_DiscursivaEspecifico INT
}

Table TBNota{
  ID_Nota INT [pk, increment, unique] // auto-increment
  NR_BrutaProva decimal
  NR_BrutaGeral decimal
  NR_BrutaObjetivaGeral decimal
  NR_BrutaDiscursivaGeral decimal
  NR_Q1_Geral decimal
  NR_LP_Q1_DiscursivaGeral decimal
```

```
  NR_Conteudo_Q1_DiscursivaGeral decimal
  NR_Q2_Geral decimal
  NR_LP_Q2_DiscursivaGeral decimal
  NR_Conteudo_Q2_DiscursivaGeral decimal
  NR_BrutaComponenteEspecifico decimal
  NR_BrutaObjetivaEspecifico decimal
  NR_BrutaDiscursivaEspecifico decimal
  NR_Q1_DiscursivaEspecifico decimal
  NR_Q2_DiscursivaEspecifico decimal
  NR_Q3_DiscursivaEspecifico decimal

}

Table QuestionarioProva{
  ID_QuestionarioProva INT [pk, increment, unique] // auto-increment
  RS_I1 varchar(1)
  RS_I2 varchar(1)
  RS_I3 varchar(1)
  RS_I4 varchar(1)
  RS_I5 varchar(1)
  RS_I6 varchar(1)
  RS_I7 varchar(1)
  RS_I8 varchar(1)
  RS_I9 varchar(1)
}

Table TBUF{
  ID_UF INT [pk, increment, unique] // auto-increment
  DS_UF varchar(3)
}

Table QuestionarioEstudante{
  ID_QuestionarioEstudante INT [pk, increment, unique] // auto-increment
  QE_I01 varchar(1)
  QE_I02 varchar(1)
  QE_I03 varchar(1)
  QE_I04 varchar(1)
  QE_I05 varchar(1)
  QE_I06 varchar(1)
  QE_I07 varchar(1)
  QE_I08 varchar(1)
  QE_I09 varchar(1)
  QE_I10 varchar(1)
  QE_I11 varchar(1)
```

```
  QE_I12 varchar(1)
  QE_I13 varchar(1)
  QE_I14 varchar(1)
  QE_I15 varchar(1)
  QE_I16 varchar(1)
  QE_I17 varchar(1)
  QE_I18 varchar(1)
  QE_I19 varchar(1)
  QE_I20 varchar(1)
  QE_I21 varchar(1)
  QE_I22 varchar(1)
  QE_I23 varchar(1)
  QE_I24 varchar(1)
  QE_I25 varchar(1)
  QE_I26 varchar(1)
}

Table TBMunicipio {
  Cod_Municipio INT [pk, increment, unique]
  DS_Nome varchar(200)

 }
Table TBRegiao {
  ID_Regiao int [pk, increment, unique]
  DS_Nome varchar(200)
}

Table TBGrauConcordanciaAluno{
  ID_GrauConcordanciaAluno int [pk, increment, unique]
  QE_I27 INT
  QE_I28 INT
  QE_I29 INT
  QE_I30 INT
  QE_I31 INT
  QE_I32 INT
  QE_I33 INT
  QE_I34 INT
  QE_I35 INT
  QE_I36 INT
  QE_I37 INT
  QE_I38 INT
  QE_I39 INT
  QE_I40 INT
  QE_I41 INT
```

```
  QE_I42 INT
  QE_I43 INT
  QE_I44 INT
  QE_I45 INT
  QE_I46 INT
  QE_I47 INT
  QE_I48 INT
  QE_I49 INT
  QE_I50 INT
  QE_I51 INT
  QE_I52 INT
  QE_I53 INT
  QE_I54 INT
  QE_I55 INT
  QE_I56 INT
  QE_I57 INT
  QE_I58 INT
  QE_I59 INT
  QE_I60 INT
  QE_I61 INT
  QE_I62 INT
  QE_I63 INT
  QE_I64 INT
  QE_I65 INT
  QE_I66 INT
  QE_I67 INT
  QE_I68 INT
}


Table Dados{
  ID_Dados INT [pk, increment, unique] // auto-increment
  NR_Ano INT
  FK_SQ_Instituicao INT
  FK_SQ_Municipio INT
  FK_SQ_UF INT
  FK_SQ_Regiao INT
  FK_SQ_TipoPresenca INT
  FK_SQ_Estudante INT
  FK_SQ_Nota INT
  FK_SQ_QuestionarioProva INT
  FK_SQ_QuestionarioEstudante INT
  FK_SQ_GrauConcordanciaAluno INT
}
```

Ref: "Dados"."FK_SQ_Instituicao" - "TBInstituicao"."ID_Instituicao"
Ref: "TBRegiao"."ID_Regiao" < "Dados"."FK_SQ_Regiao"
Ref: "TBMunicipio"."Cod_Municipio" < "Dados"."FK_SQ_Municipio"
Ref: "TBTipoPresenca"."ID_TipoPresenca" < "Dados"."FK_SQ_TipoPresenca"
Ref: "TBEstudante"."ID_Estudante" < "Dados"."FK_SQ_Estudante"
Ref: "TBNota"."ID_Nota" < "Dados"."FK_SQ_Nota"
Ref: "QuestionarioEstudante"."ID_QuestionarioEstudante" < "Dados"."FK_SQ_QuestionarioEstudante"
Ref: "QuestionarioProva"."ID_QuestionarioProva" < "Dados"."FK_SQ_QuestionarioProva"
Ref: "Dados"."FK_SQ_GrauConcordanciaAluno" - "TBGrauConcordanciaAluno"."ID_GrauConcordanciaAluno"
Ref: "Dados"."FK_SQ_UF" - "TBUF"."ID_UF"

Ref: "Dados"."FK_SQ_Instituicao" < "Dados"."NR_Ano"

[2] ***Script SQL***

[https://github.com/acmont/Prova-individual---Data-Warehouse-21.1/blob/main/scripts%20banco/individual_1.sql]

[https://github.com/acmont/Prova-individual---Data-Warehouse-21.1/blob/main/scripts%20banco/individual_2.sql]