

Universidade Federal do Rio de Janeiro
Instituto de Computação
Professor: Geraldo Xexéo

Prova de Dw - Análise dos Dados do ENADE

Andreza Cardoso Santos

https://github.com/AndrezaCS/Prova_Individual_DW

Novembro 2021

Observações: Estou enviando o workflow do Knime por email (amoraacs@gmail.com) hoje, dia 29/11/2020, pois mesmo comprimindo o arquivo, não consigo adicionar ele ao meu repositório do GitHub por causa do tamanho do documento. Para a realização do trabalho, segui o mesmo método utilizado no trabalho em grupo, mas tive dificuldade, pois como meu laptop possui apenas 4gb de RAM, meu computador travava muito por causa do Knime, levando a CPU, Memória e Disco a 100% algumas vezes.

Questão 1)

No site do ENADE, <https://www.gov.br/inep/pt-br/aceso-a-informacao/dados-abertos/microdados/enade>, podemos encontrar na área de microdados a pasta zipada com os seguintes dados referentes a cada ano (2017-2019):

	2017	2018	2019
LEIA-ME	- Manual do usuário 2017.pdf - Questionário do Estudante do Enade_Edição 2017.pdf - Questionário do Estudante do Enade_Compl Licenciatura_Edição 2017.pdf - Dicionário de variáveis dos Microdados do Enade_Edição 2017.ods - Dicionário de variáveis dos Microdados do Enade_Edição 2017.xlsx	-Manual do usuário 2018.pdf; -Questionário do Estudante_Enade_Edição 2018.pdf; -Dicionário de Variáveis dos Microdados do Enade_Edição 2018.ods; -Dicionário de Variáveis dos Microdados do Enade_Edição 2018.xlsx;	- Manual do usuário 2019.pdf; - Questionário do Estudante_Enade_Edição 2019.pdf; - Dicionário de Variáveis dos Microdados do Enade_Edição 2019.ods; - Dicionário de Variáveis dos Microdados do Enade_Edição 2019.xlsx;
INPUTS	-sas_input_enade_2017.sas -R_input_enade_2017.R -spss_input_enade_2017.sps	-sas_input_enade_2018.sas; -R_input_enade_2018.R; -spss_input_enade_2018; ;	-sas_input_enade_2019.sas; -R_input_enade_2019.r; -spss_input_enade_2019.spss;

DADOS	-microdados_enade_2017.txt	-microdados_enade_2018.txt.	-microdados_enade_2019.txt.
-------	----------------------------	-----------------------------	-----------------------------

Tabela 1: Dados das pastas

Para fazer a extração desses dados para a ferramenta *KNIME*, precisei utilizar dois métodos, pois tive problema com o link. Primeiro, para o ano de 2017, extrai os dados diretamente do link do ENADE, depois, para os dois anos seguintes, coloquei os dados em *.txt* no meu próprio *GitHub* e fiz a conexão com o *KNIME* diretamente de lá, bem como a sugestão utilizada pelo professor, mas no qual o link também parou de funcionar, então apenas segui a mesma ideia.

Para isso, foi chamada a função “*Unzip files (legacy)*” para baixar o arquivo e descompactar automaticamente salvando em uma pasta local do Knime. Os dados foram coletados a partir dos seguintes links:

- https://download.inep.gov.br/microdados/Enade_Microdados/microdados_Enade_2017_portal_2018.10.09.zip
- https://github.com/AndrezaCS/Prova_Individual_DW/raw/main/Dados%20ENADE/microdados_enade_2018.zip
- https://github.com/AndrezaCS/Prova_Individual_DW/raw/main/Dados%20ENADE/microdados_enade_2019.zip

Abaixo, segue a imagem dos nós utilizados no *Knime* na parte de extração.

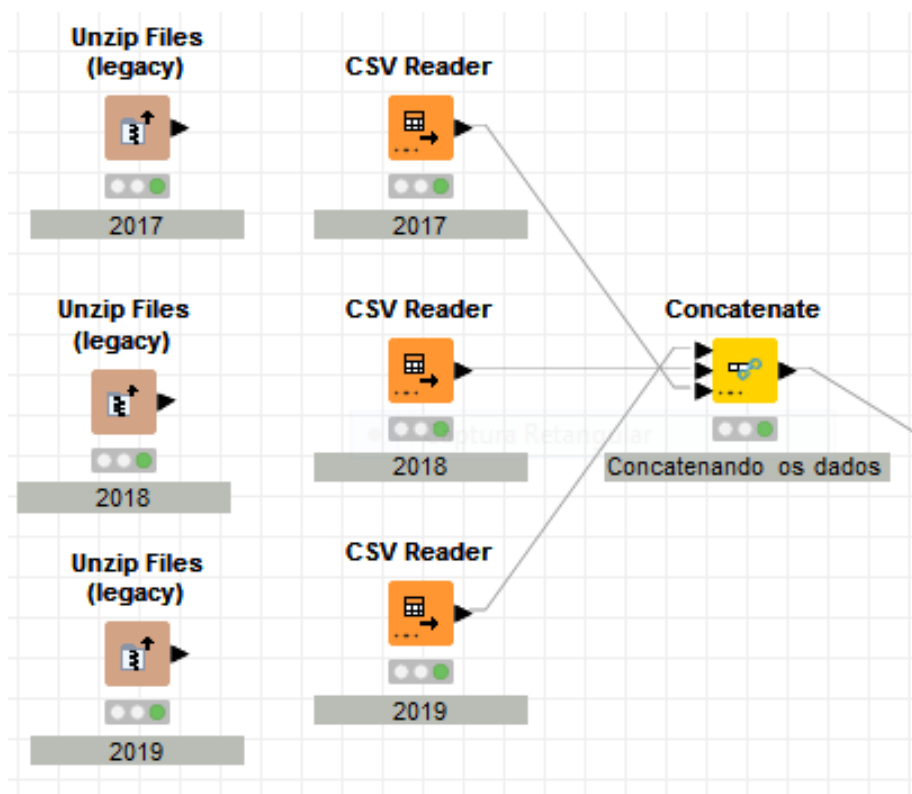


Figura 1: Extração dos dados direto pelo *Knime*

Depois configurei os nós “Unzip files (legacy)” para cada ano, como mostrado nas figuras abaixo:

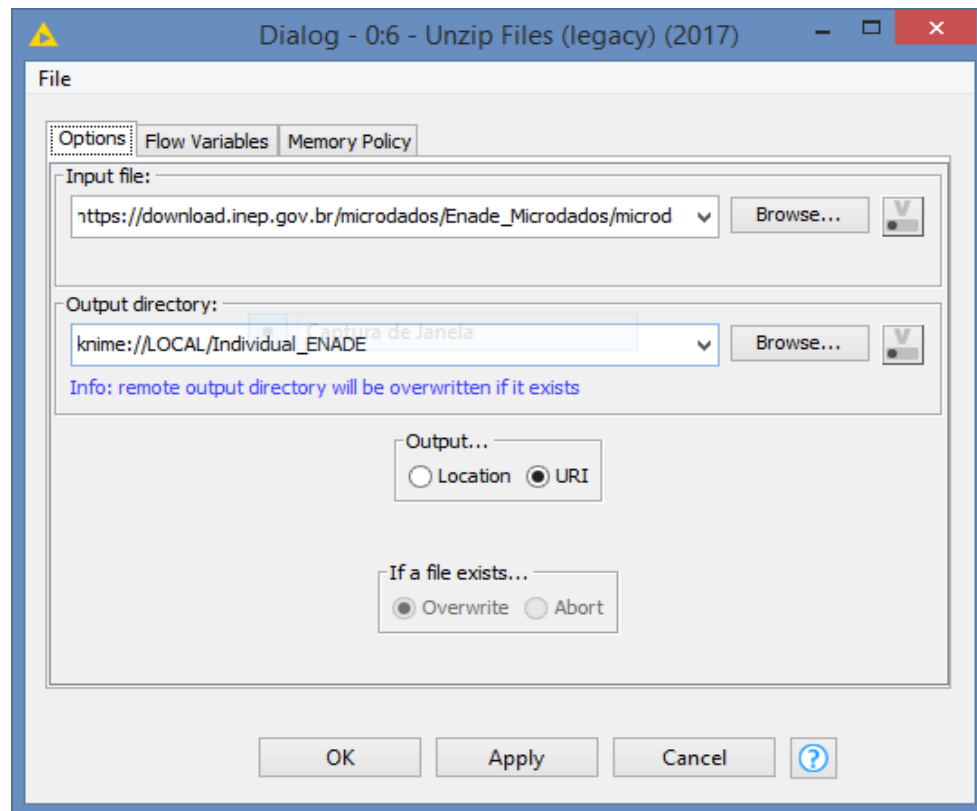


Figura 2 - Configuração do “Unzip files (legacy)” para 2017

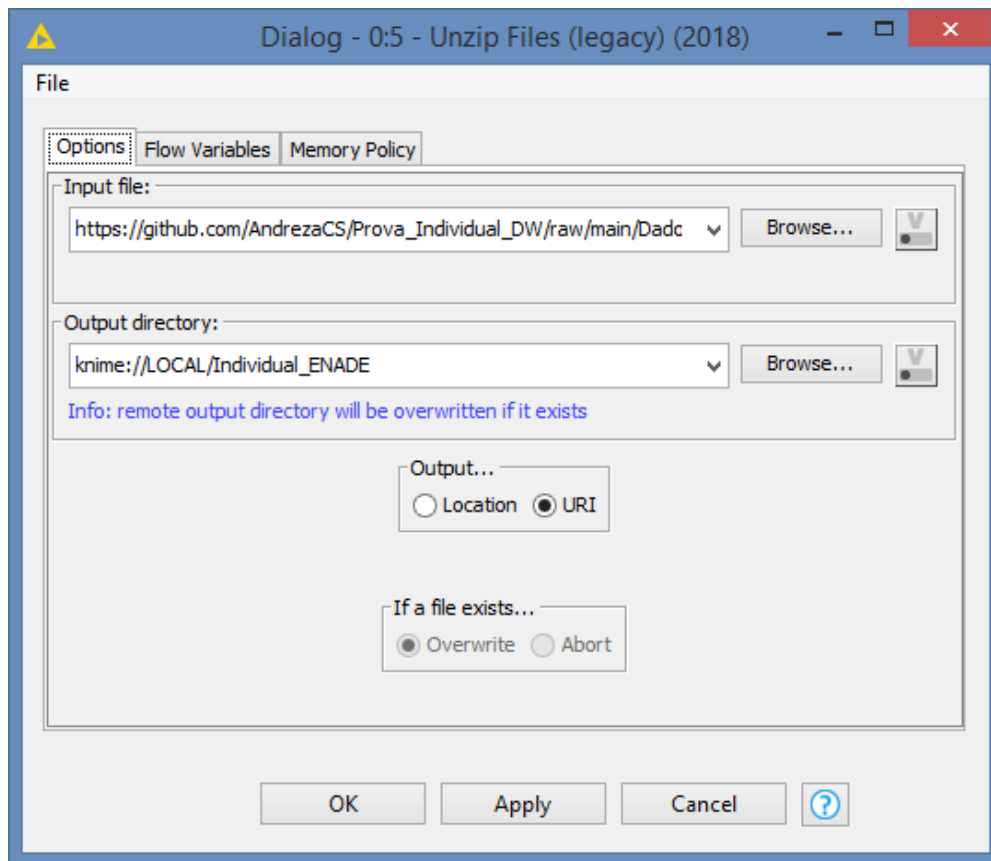


Figura 3 - Configuração do “Unzip files (legacy)” para 2019

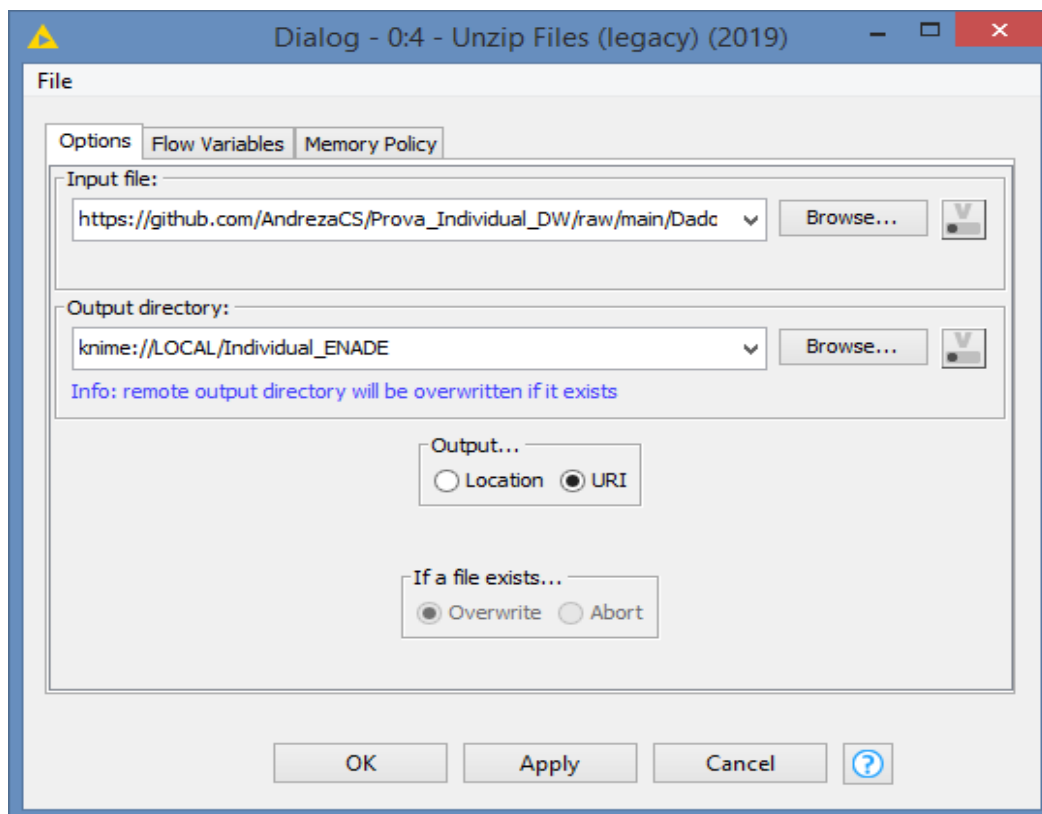


Figura 4 - Configuração do “Unzip files (legacy)” para 2019

Após esses passos utilizei o nó “CSV Reader” para ler os arquivos em CSV descompactados anteriormente, para que assim fosse possível analisar os dados obtidos. Para que o arquivo seja lido sem precisar procurar o arquivo, ajustei a configuração do CSV Reader para o modo 'Mountpoint'/ 'LOCAL' e no campo 'Format' seleciona-se o 'Autodetect format' para que seja feita a separação correta do arquivo.

Abaixo seguem as imagens das configurações utilizadas.

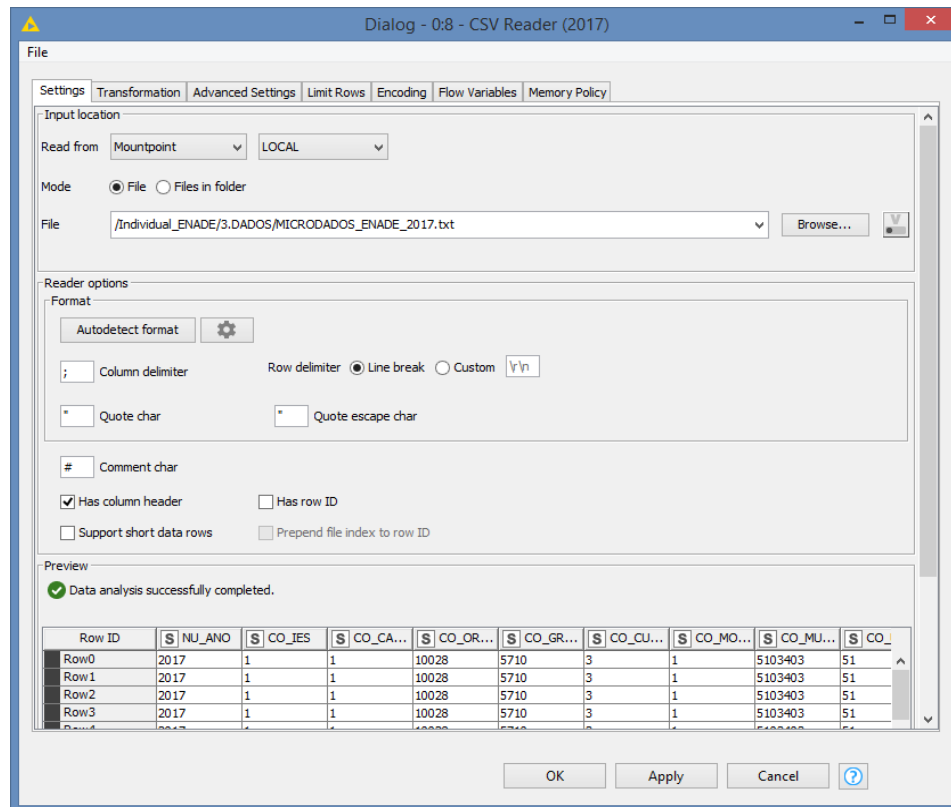


Figura 5 - Configuração do “CSV Reader”

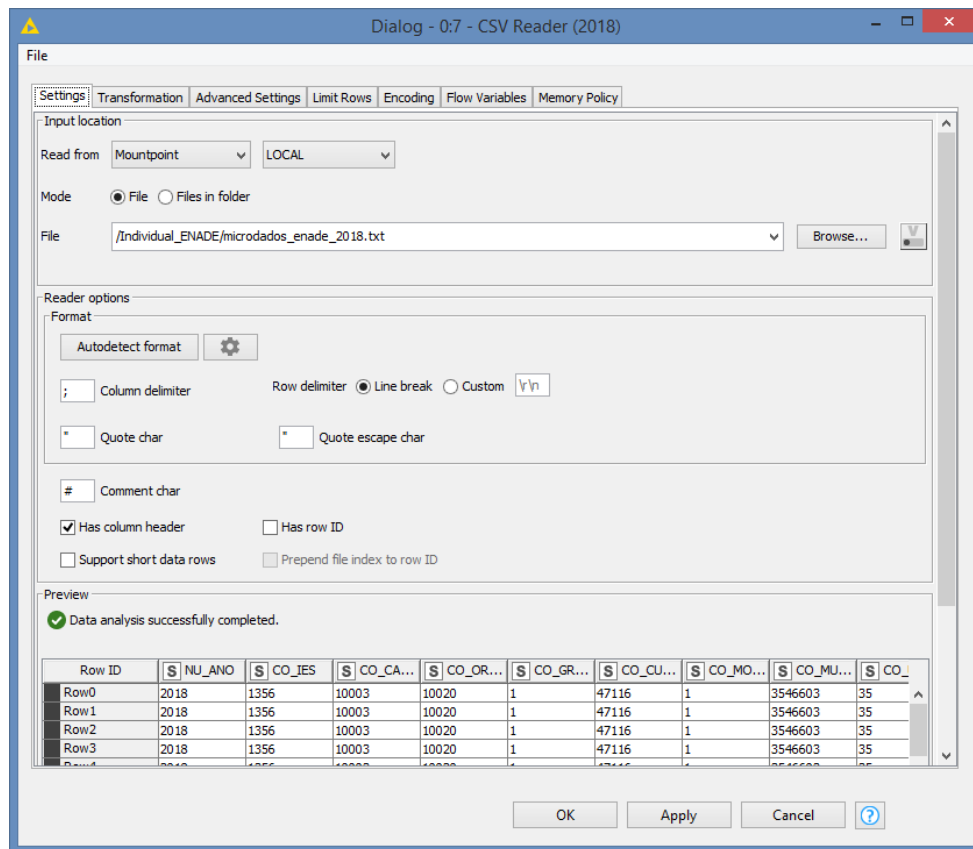


Figura 6 - Configuração do “CSV Reader”

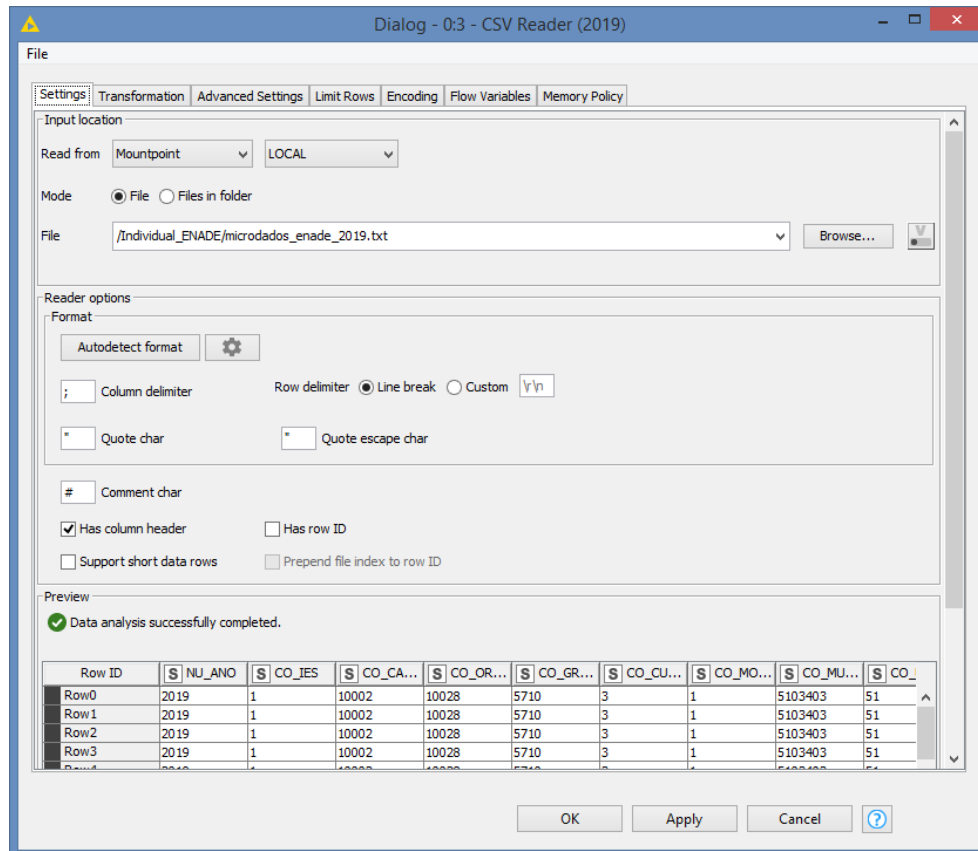


Figura 7 - Configuração do “CSV Reader”

Depois de feito isso, usei o nó “Concatenate” para unir os três CSV, para facilitar a carga.

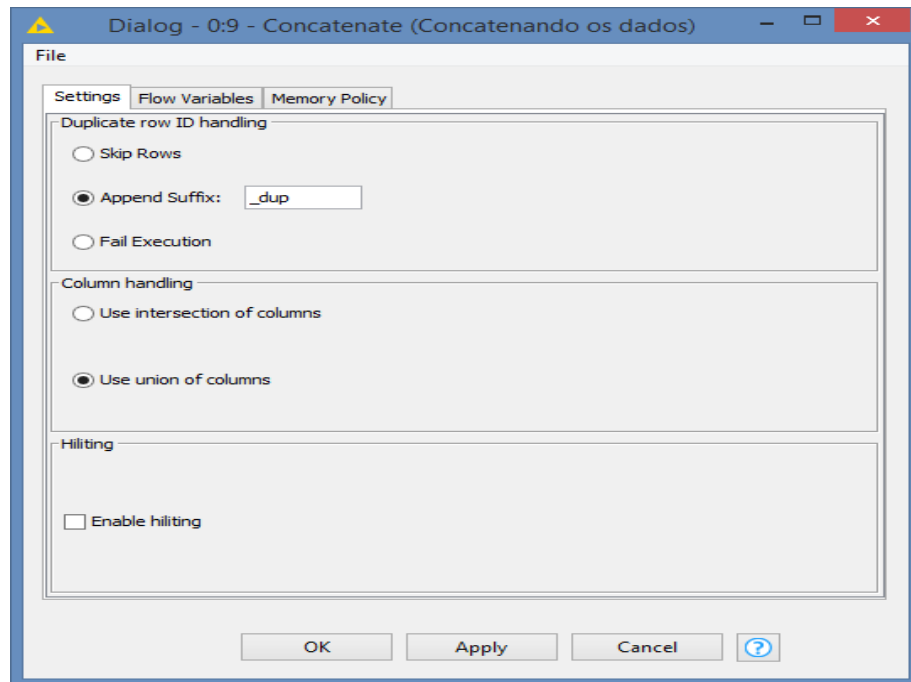


Figura 8 - Configuração do “Concatenate”

Questão 2)

Criei o modelo dimensional (estrela) a partir dos dados baixados do dicionário de dados desses 3 anos (2017-2019) na seguinte ferramenta: *Db Diagram*:

(<https://dbdiagram.io/d/6181a30ad5d522682df5b7aa>).

Fiz apenas um modelo, pois os dados dos dicionários eram os mesmos em relação aos 3 anos, só diferenciavam algumas pequenas questões que não influenciaram no modelo a ser criado. Abaixo segue a imagem do modelo estrela gerado, com uma tabela fato “Dados” e 10 (dez) dimensões, que são: “TBInstituicao”, “TBEstudante”, “TBTipoPresenca”, “TBNota”, “QuestionarioProva”, “TBUF”, “ QuestionarioEstudante”, “TBMunicipio”, “TBRegiao”, “TBGrauConcordanciaAluno”.

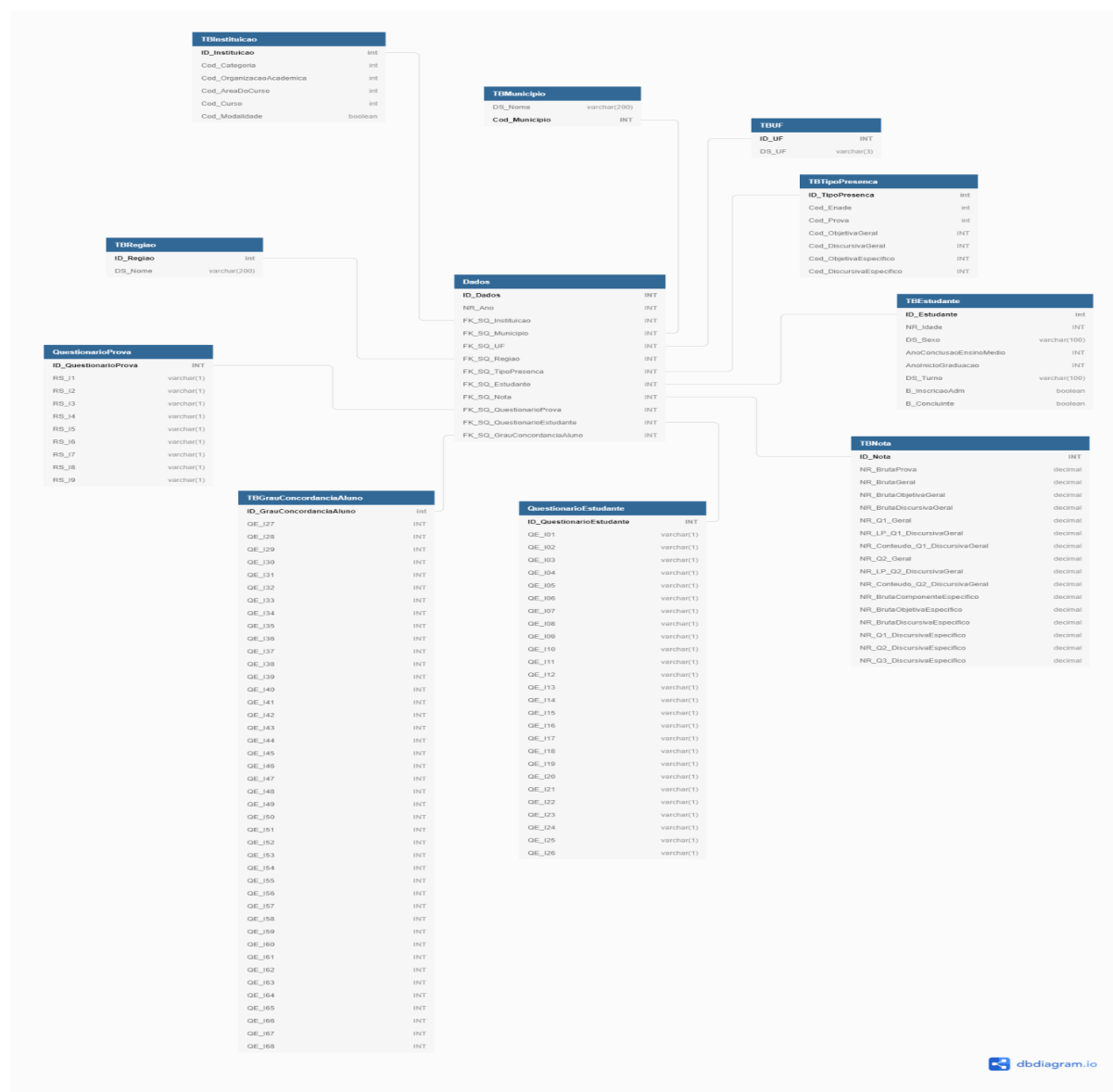


Figura 9 - Modelo Estrela

Código: No ANEXO [1] Script para o Modelo Estrela

Questão 3)

Para essa questão, usei um servidor externo para criar a base de dados no MySQL versão 5.7 para suportar a carga de dados. Tive que usar a versão 5.7, porque um nó usado no *Knime* para carregar os dados não suportava o *MySQL* com versão maior, devido o nó estar obsoleto, mesmo problema encontrado no trabalho em grupo. Usei a ferramenta *DataGrip* para criar o banco de dados relacional de acordo com o modelo construído acima (Figura 9), no script do Modelo Estrela e então gerei o *script* da estrutura do banco que se encontra no Anexo [2] Script SQL, que contém o Script 1, a Tabela Auxiliar e a Trigger contidas no meu repositório do GitHub.

Questão 4)

Segui o mesmo raciocínio utilizado para os dados da Covid no trabalho em grupo, ou seja, criei um fluxo no *KNIME* para que a carga de dados fosse feita de uma vez só, desde o download dos arquivos até o tratamento dos dados, como mostra a figura abaixo:

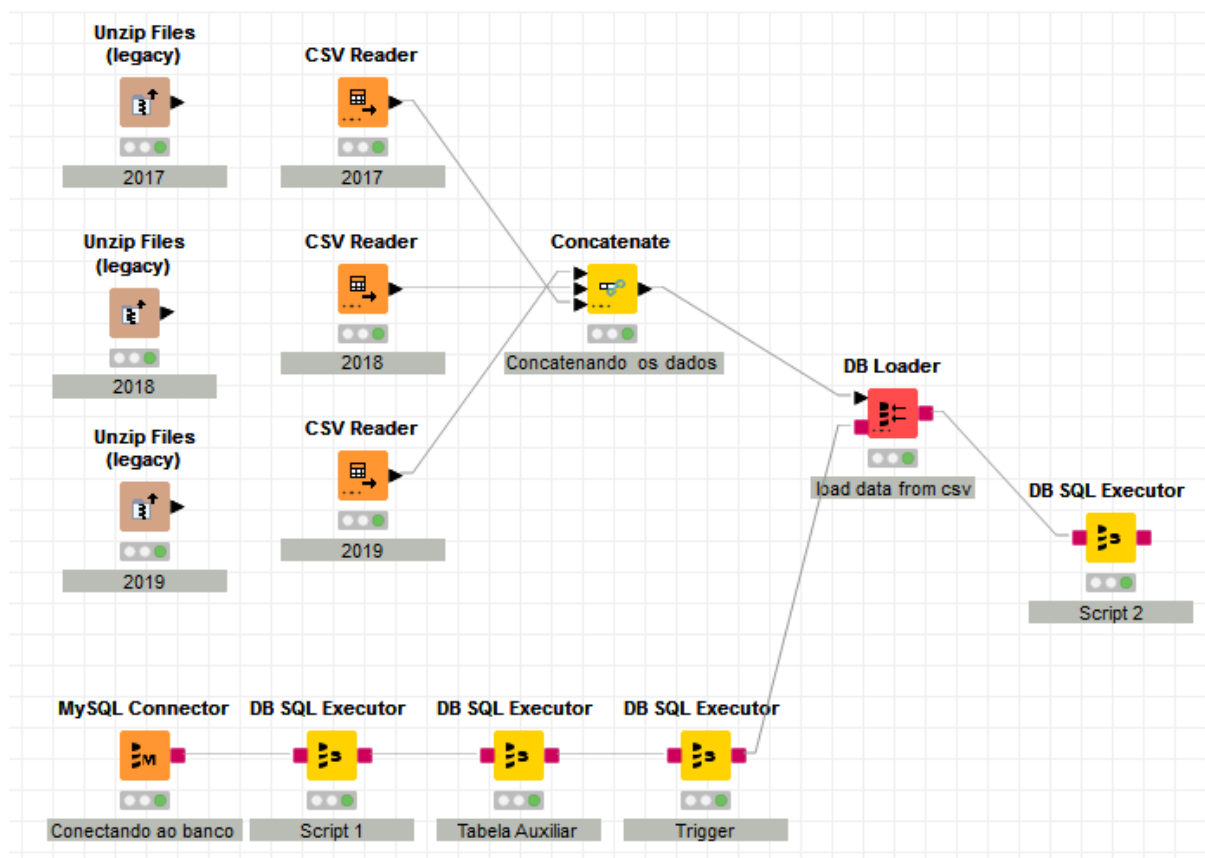


Figura 10 - Fluxo no Knime para a carga dos dados.

Descrição dos nós utilizados no fluxo:

- MySQL Connector (Conectando ao banco): Primeiro adicionei o nó '*MySQL Connector*' onde coloquei as configurações de conexão do banco local, sendo elas: IP, porta, usuário e senha;
- DB SQL Executor (Script 1): Nesse nó foi executado o *script* do anexo 2.2.1, onde foi criada o *database*: DW_Individual_Andreza;
- DB SQL Executor (Tabela Auxiliar): Nesse nó foi executado o *script* do anexo 2.2.2, com a criação da tabela_aux (Figura 11) para ajudar no tratamento dos dados, com todos eles em varchar(100);

Figura 11 - Tabela Auxiliar (Figuras abaixo)

▼	tabela_aux
▼	columns 151
	NU_ANO varchar(100)
	CO_CATEGAD varchar(100)
	CO_ORGACAD varchar(100)
	CO_IES varchar(100)
	CO_GRUPO varchar(100)
	CO_CURSO varchar(100)
	CO_MODALIDADE varchar(100)
	CO_MUNIC_CURSO varchar(100)
	CO_UF_CURSO varchar(100)
	CO_REGIAO_CURSO varchar(100)
	NU_IDADE varchar(100)
	TP_SEXO varchar(100)
	ANO_FIM_EM varchar(100)
	ANO_IN_GRAD varchar(100)
	CO_TURNO_GRADUACAO varchar(100)
	TP_INSCRICAO_ADM varchar(100)
	TP_INSCRICAO varchar(100)
	NU_ITEM_OFG varchar(100)
	NU_ITEM_OFG_Z varchar(100)
	NU_ITEM_OFG_X varchar(100)
	NU_ITEM_OFG_N varchar(100)
	NU_ITEM_OCE varchar(100)

	NU_ITEM_OCE_Z varchar(100)
	NU_ITEM_OCE_X varchar(100)
	NU_ITEM_OCE_N varchar(100)
	DS_VT_GAB_OFG_ORIG varchar(100)
	DS_VT_GAB_OFG_FIN varchar(100)
	DS_VT_GAB_OCE_ORIG varchar(100)
	DS_VT_GAB_OCE_FIN varchar(100)
	DS_VT_ESC_OCE varchar(100)
	DS_VT_ACE_OCE varchar(100)
	TP_PRES varchar(100)
	TP_PR_GER varchar(100)
	DS_VT_ESC_OFG varchar(100)
	DS_VT_ACE_OFG varchar(100)
	TP_PR_OB_FG varchar(100)
	TP_PR_DI_FG varchar(100)
	TP_PR_OB_CE varchar(100)
	TP_PR_DI_CE varchar(100)
	TP_SFG_D1 varchar(100)
	TP_SFG_D2 varchar(100)
	TP_SCE_D1 varchar(100)
	TP_SCE_D2 varchar(100)
	TP_SCE_D3 varchar(100)
	NT_GER varchar(100)
	NT_FG varchar(100)

	NT_OBJ_FG varchar(100)
	NT_DIS_FG varchar(100)
	NT_FG_D1 varchar(100)
	NT_FG_D1_PT varchar(100)
	NT_FG_D1_CT varchar(100)
	NT_FG_D2 varchar(100)
	NT_FG_D2_PT varchar(100)
	NT_FG_D2_CT varchar(100)
	NT_CE varchar(100)
	NT_OBJ_CE varchar(100)
	NT_DIS_CE varchar(100)
	NT_CE_D1 varchar(100)
	NT_CE_D2 varchar(100)
	NT_CE_D3 varchar(100)
	CO_RS_I1 varchar(100)
	CO_RS_I2 varchar(100)
	CO_RS_I3 varchar(100)
	CO_RS_I4 varchar(100)
	CO_RS_I5 varchar(100)
	CO_RS_I6 varchar(100)
	CO_RS_I7 varchar(100)
	CO_RS_I8 varchar(100)
	CO_RS_I9 varchar(100)
	QE_I01 varchar(100)

	QE_I02 varchar(100)
	QE_I03 varchar(100)
	QE_I04 varchar(100)
	QE_I05 varchar(100)
	QE_I06 varchar(100)
	QE_I07 varchar(100)
	QE_I08 varchar(100)
	QE_I09 varchar(100)
	QE_I10 varchar(100)
	QE_I11 varchar(100)
	QE_I12 varchar(100)
	QE_I13 varchar(100)
	QE_I14 varchar(100)
	QE_I15 varchar(100)
	QE_I16 varchar(100)
	QE_I17 varchar(100)
	QE_I18 varchar(100)
	QE_I19 varchar(100)
	QE_I20 varchar(100)
	QE_I21 varchar(100)
	QE_I22 varchar(100)
	QE_I23 varchar(100)
	QE_I24 varchar(100)
	QE_I25 varchar(100)

```

QE_I26 varchar(100)
QE_I27 varchar(100)
QE_I28 varchar(100)
QE_I29 varchar(100)
QE_I30 varchar(100)
QE_I31 varchar(100)
QE_I32 varchar(100)
QE_I33 varchar(100)
QE_I34 varchar(100)
QE_I35 varchar(100)
QE_I36 varchar(100)
QE_I37 varchar(100)
QE_I38 varchar(100)
QE_I39 varchar(100)
QE_I40 varchar(100)
QE_I41 varchar(100)
QE_I42 varchar(100)
QE_I43 varchar(100)
QE_I44 varchar(100)
QE_I45 varchar(100)
QE_I46 varchar(100)
QE_I47 varchar(100)
QE_I48 varchar(100)
QE_I49 varchar(100)

```

```

QE_I74 varchar(100)
QE_I75 varchar(100)
QE_I76 varchar(100)
QE_I77 varchar(100)
QE_I78 varchar(100)
QE_I79 varchar(100)
QE_I80 varchar(100)
QE_I81 varchar(100)
ID_AUX int(11) (auto increment)

```

```

QE_I50 varchar(100)
QE_I51 varchar(100)
QE_I52 varchar(100)
QE_I53 varchar(100)
QE_I54 varchar(100)
QE_I55 varchar(100)
QE_I56 varchar(100)
QE_I57 varchar(100)
QE_I58 varchar(100)
QE_I59 varchar(100)
QE_I60 varchar(100)
QE_I61 varchar(100)
QE_I62 varchar(100)
QE_I63 varchar(100)
QE_I64 varchar(100)
QE_I65 varchar(100)
QE_I66 varchar(100)
QE_I67 varchar(100)
QE_I68 varchar(100)
QE_I69 varchar(100)
QE_I70 varchar(100)
QE_I71 varchar(100)
QE_I72 varchar(100)
QE_I73 varchar(100)

```

- DB SQL Executor (Trigger): Nesse nó foi executado o *script* do anexo 2.2.3, onde a *Trigger* é utilizada para auxiliar na inserção de dados;
- DB Loader (load data from csv): Este nó faz a carga dos dados para o banco local, onde configurei o banco de dados e a tabela_aux. A estrutura da tabela auxiliar está igual ao arquivo CSV, contendo a mesma ordem das colunas e todos os campos são *VARCHAR*, pois assim os dados são melhor otimizados.

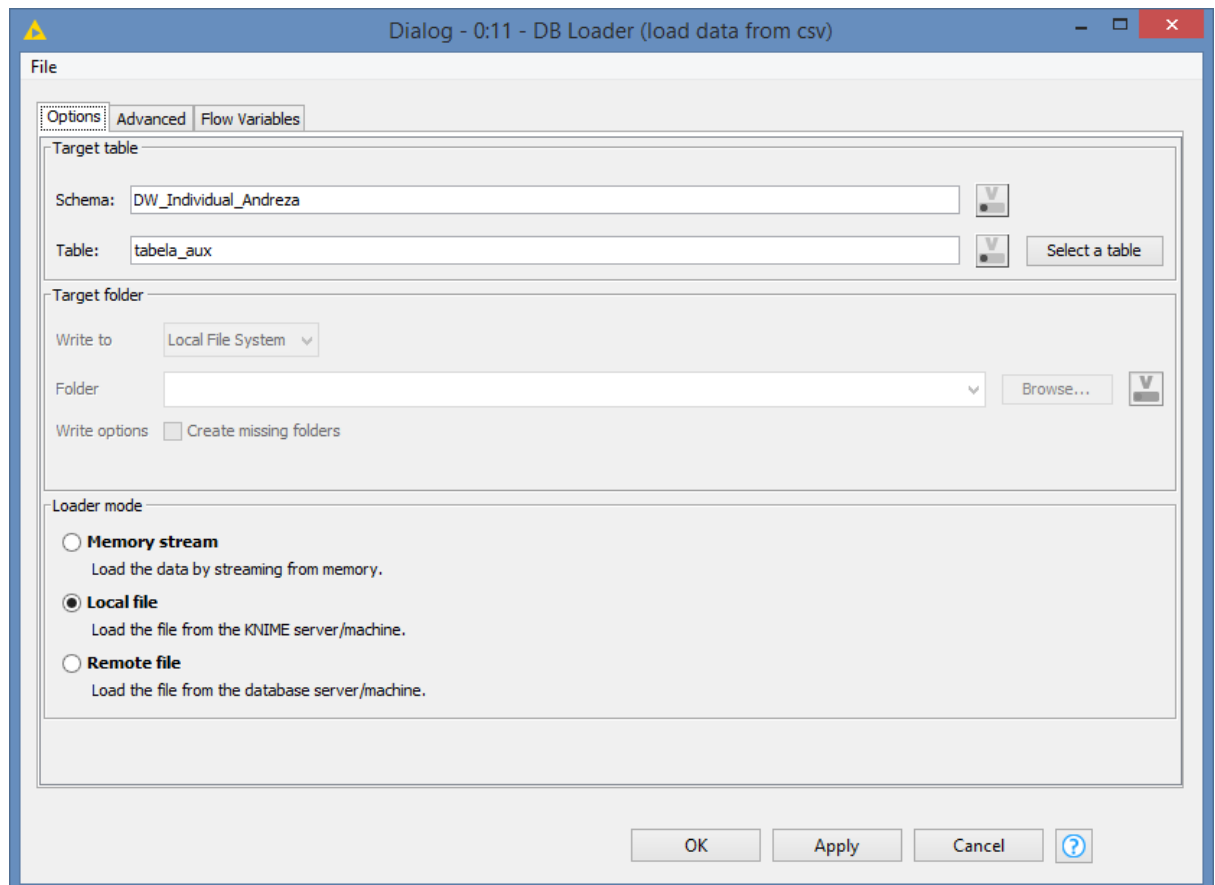


Figura 12 - Configuração do DB Loader

- DB SQL Executor (Script 2): Início do tratamento de dados, onde tratei os dados vazios como nulos. Como mostrado no Anexo no Script [3] Carga de dados

Questão 5)

Nessa questão, utilizei a ferramenta *Tableau* para a realização dos gráficos e tabelas. O banco de dados das questões anteriores foi conectado ao Tableau para que fosse possível utilizar estes dados e dessa forma fazer as análises. As tabelas referentes a cada pergunta encontram-se no repositório do GitHub.

Elaborei as seguintes 5 questões para analisar os dados do ENADE, são elas:

- Pergunta 1: Qual a comparação dos inscritos do tipo concluinte por ano?

Concluinte por Ano

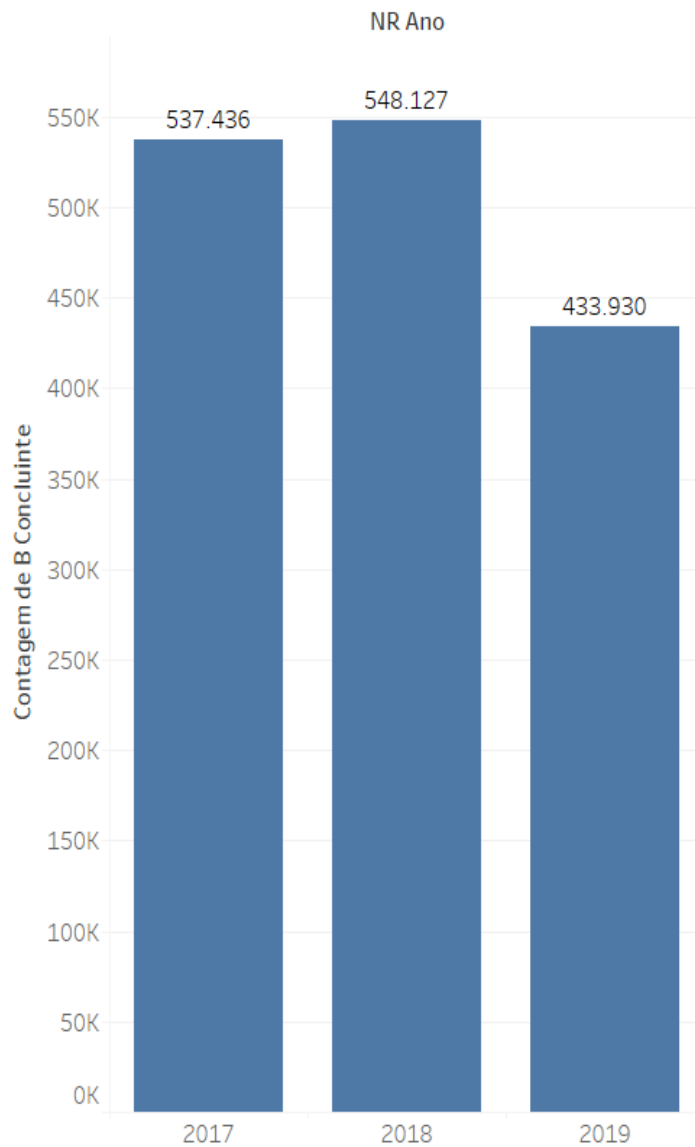


Figura 12 - Gráfico Concluinte por ano

Pelo gráfico acima, podemos perceber que o ano de 2018, teve um maior índice de alunos concluintes e que o ano de 2019 houve uma certa queda em relação aos dois anos anteriores.

- Pergunta 2: Qual a comparação dos gêneros que fizeram as provas nos 3 anos?

Genero por ano

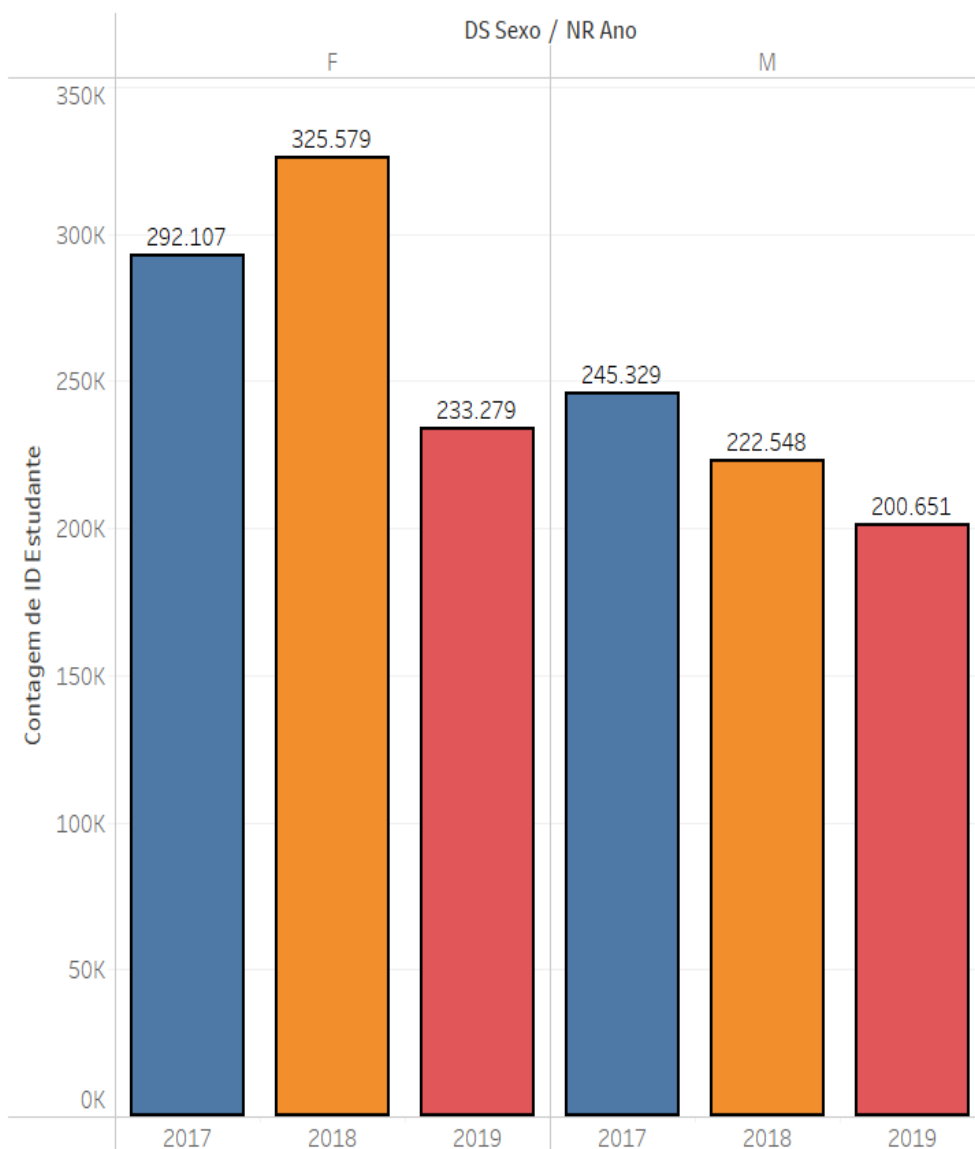


Figura 13 - Gráfico de Gênero por ano

Através da comparação dos gêneros no gráfico, pode-se perceber um número bem mais expressivo em relação às pessoas do sexo feminino fazendo a prova, em relação às do sexo masculino.

- Pergunta 3: Qual a comparação dos 3 anos sobre a região com mais inscritos?



Figura 14 - Gráfico Inscritos por região

É possível observar um número muito superior de inscritos na região Sudeste em comparação com as demais.

- Pergunta 4: Qual gênero por categoria mais fez a prova no ano de 2017?

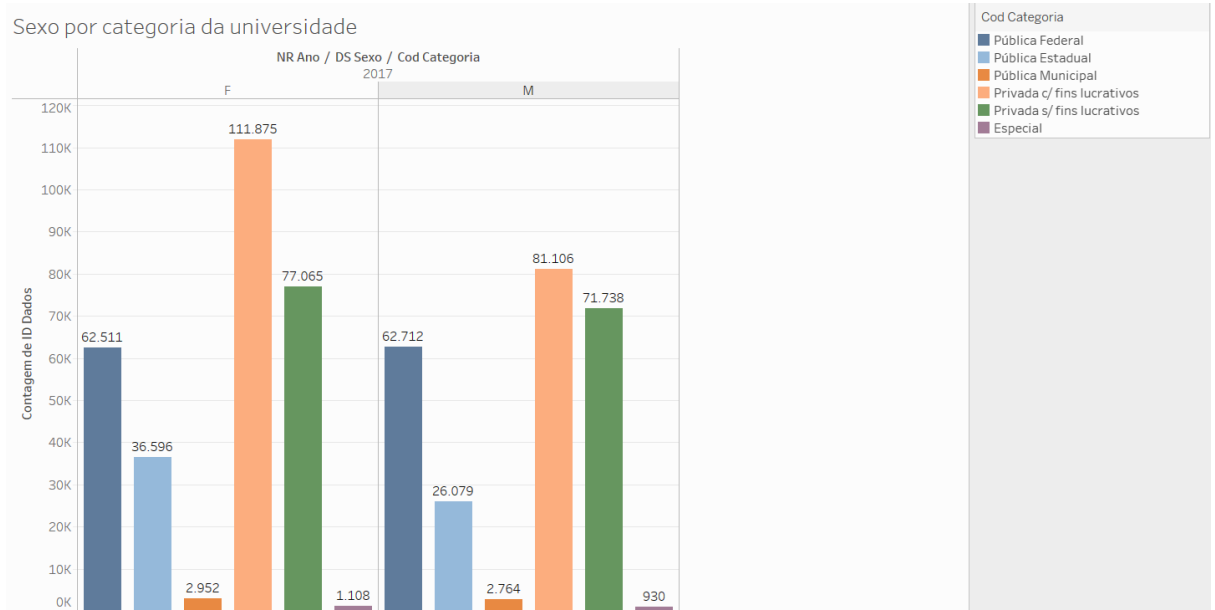


Figura 15 - Gráfico de Sexo por categoria de universidade em 2017

Observando o gráfico acima é possível perceber que houve uma adesão muito maior do gênero feminino em todas as categorias, com exceção da Pública Federal no ano de 2017.

- Pergunta 5: Qual turno mais fez a prova?

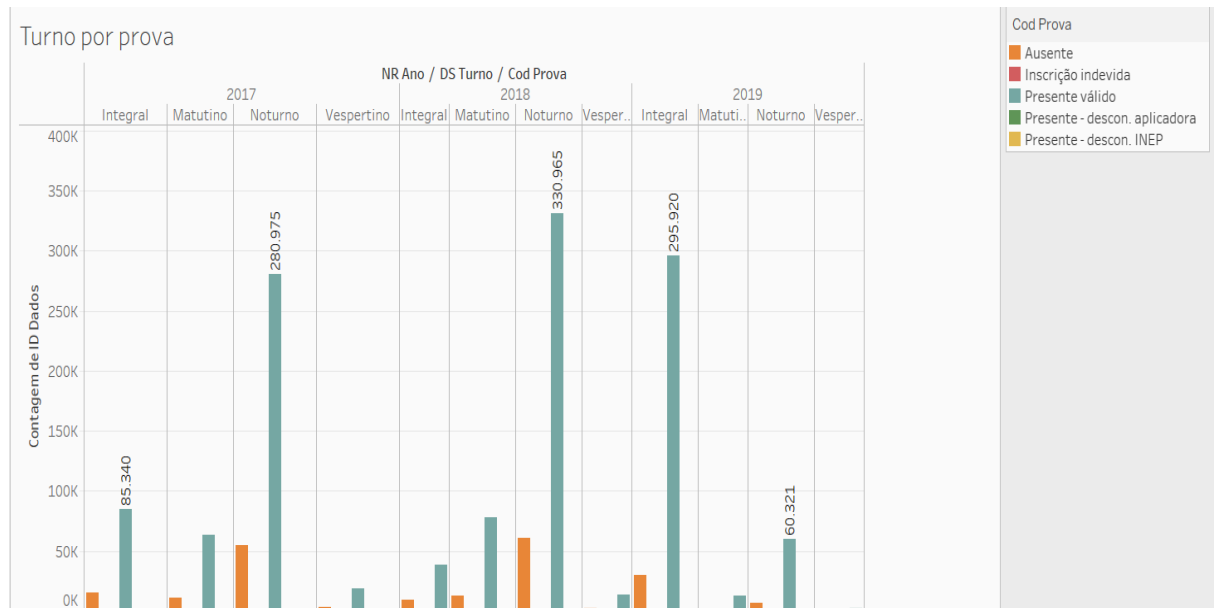


Figura 16 - Gráfico Turno por prova

Nota-se que o turno noturno abrange a maior parte dos alunos que fizeram as provas do ENADE em todos os anos, com um valor muito superior em relação aos demais turnos.

Questão 6)

Nessa questão, segui o mesmo fluxo feito no trabalho em grupo, apenas conectando aos dados do Enade, e tentando trazer a ideia de algum aprendizado, porém, o fluxo até rodou, mas a acurácia e precisão não chegavam ao satisfatório de 60% como solicitado, os valores que encontrei foram muito baixos, então, segue na imagem abaixo o fluxo tentado.

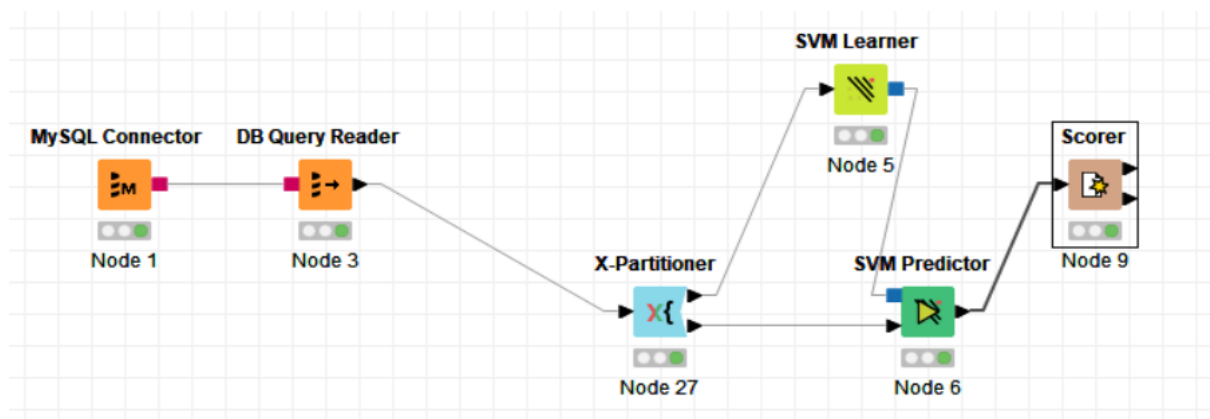


Figura 17 - Fluxo do Knime para o aprendizado

Questão 7)

- Db Diagram: <https://dbdiagram.io/d/6181a30ad5d522682df5b7aa> - Essa ferramenta foi escolhida por conta da sua facilidade em montar o modelo dimensional.
- DataGrip: <https://www.jetbrains.com/datagrip/> - Essa ferramenta foi escolhida por ser mais intuitiva que o MySQL workbench.
- Knime: <https://www.knime.com/> - Essa ferramenta foi utilizada para fazer o download automático dos dados e para o aprendizado de máquina.
- Tableau: <https://www.tableau.com/pt-br> - Essa ferramenta é muito utilizada para a geração de gráficos e tabelas.
- GitHub: https://github.com/AndrezaCS/Prova_Individual_DW - O GitHub foi utilizado para colocar o repositório dos dados analisados.
- Google drive: Para o envio do workflow do Knime

ANEXO

[1] Script para o Modelo Estrela

//// -- Tables and References

```
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_Concluente 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 {
  DS_Nome varchar(200)
  Cod_Municipio INT [pk, increment,unique]
}

```

```

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

```

```
Table TBGruaConcordanciaAluno{
  ID_GruaConcordanciaAluno 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"
```

[2] Script SQL

[2.2.1]

https://github.com/AndrezaCS/Prova_Individual_DW/blob/main/Quest%C3%A3o%203%20e%20Quest%C3%A3o%204/Script%201.sql

[2.2.2]

https://github.com/AndrezaCS/Prova_Individual_DW/blob/main/Quest%C3%A3o%203%20e%20Quest%C3%A3o%204/Tabela%20Auxiliar.sql

[2.2.3]

https://github.com/AndrezaCS/Prova_Individual_DW/blob/main/Quest%C3%A3o%203%20e%20Quest%C3%A3o%204/Trigger.sql

[3] Script Carga de Dados

https://github.com/AndrezaCS/Prova_Individual_DW/blob/main/Quest%C3%A3o%203%20e%20Quest%C3%A3o%204/Script%202.sql