PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO (PUC-RIO)
PÓS-GRADUAÇÃO EM CIÊNCIA DE DADOS E ANALYTICS
MVP – ENGENHARIA DE DADOS

ANÁLISE DE FATORES SOCIOECONÔMICOS E COMPORTAMENTAIS NO DESEMPENHO ESTUDANTIL

Hugo Henrique Barreiros Grifo Link do GitHub

Rio de Janeiro

2025

SUMÁRIO

1. Objetivo do Projeto

Este projeto tem como objetivo realizar a construção de um pipeline de Engenharia de Dados completo (end-to-end) na nuvem, utilizando a plataforma **Databricks Community Edition**. O foco principal é analisar quais fatores socioeconômicos, familiares e comportamentais exercem maior influência no desempenho acadêmico e na aprovação de estudantes.

O dataset utilizado foi o "**Student Performance**" (selecionando apenas o dataset de matemática "**student-mat**") obtido no *UCI Machine Learning Repository*. A análise busca responder a perguntas cruciais, como o impacto do consumo de álcool nas notas finais, a correlação entre a escolaridade dos pais e o sucesso do aluno, e a influência do tempo de estudo no desempenho escolar.

Repositório do Projeto:
https://github.com/HUGOGRIFOPERSONAL/HUGO_GRIFO_PUC_RIO_POS

Repositório dos Dados:
https://archive.ics.uci.edu/dataset/320/student+performance

2. Arquitetura da Solução

Para garantir escalabilidade, organização e qualidade dos dados, adotou-se a Arquitetura Medalhão (Medallion Architecture), processando os dados em três camadas distintas:

- **Camada *Bronze* (*Raw*):** Armazenamento dos dados brutos, exatamente como foram ingeridos da fonte, garantindo a preservação do histórico original.

- **Camada *Silver* (*Trusted*):** Dados limpos, padronizados e tratados. Nesta etapa, foram aplicadas renomeações de colunas para o português, tipagem correta dos dados e remoção de duplicatas.

- **Camada *Gold* (*Refined*):** Dados modelados para análise e consultas. Foi utilizado um **Esquema Estrela (*Star Schema*)**, separando os dados em tabelas Fato (métricas) e Dimensão (contexto) para otimizar a performance de consultas analíticas.

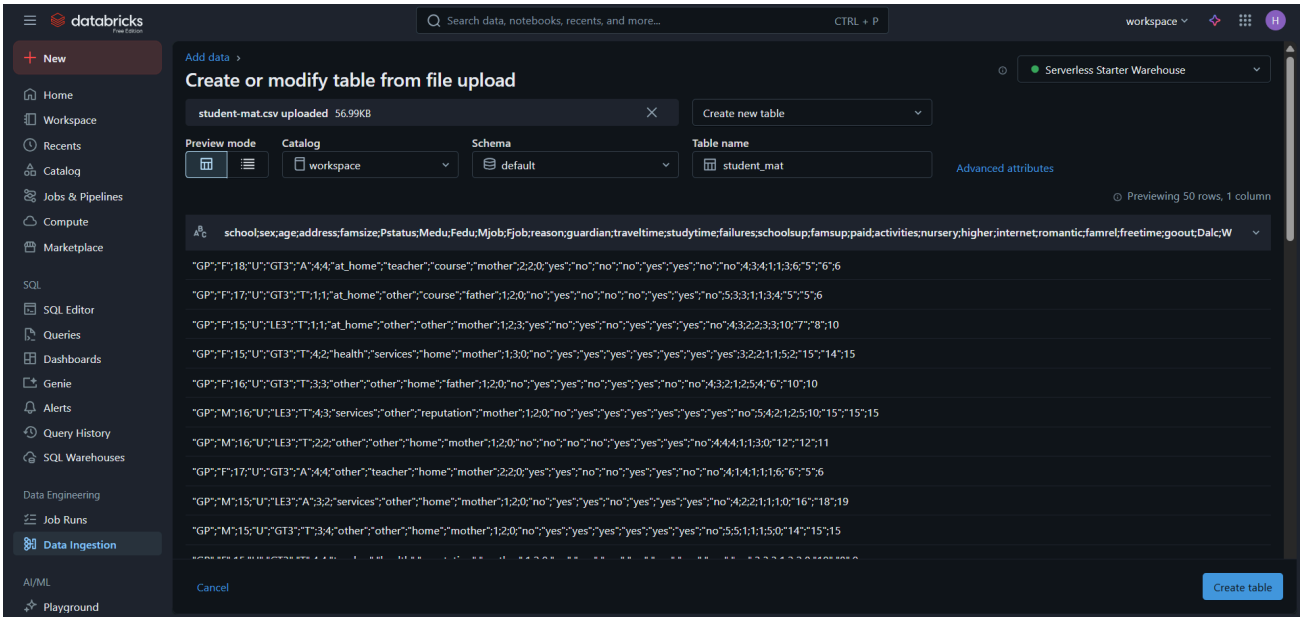
A ferramenta escolhida foi o **Databricks**, utilizando **PySpark** para o processamento massivo de dados e **Spark SQL** para análises exploratórias.

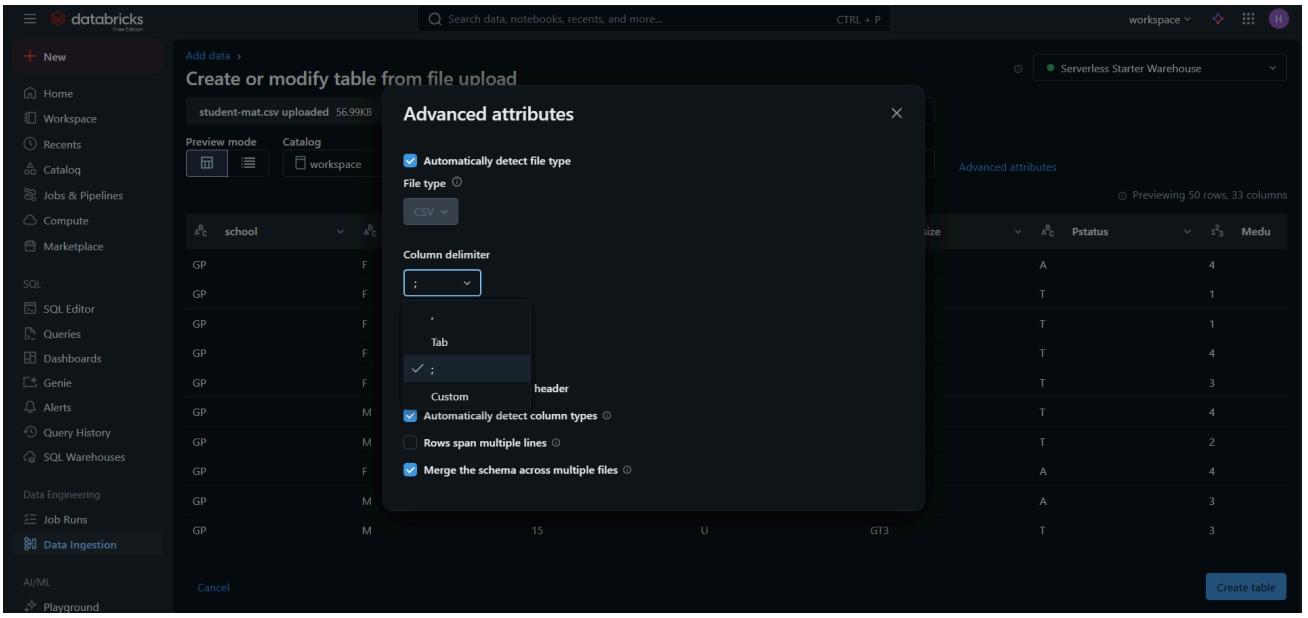
3. Coleta de Dados (Ingestão - Camada *Bronze*)

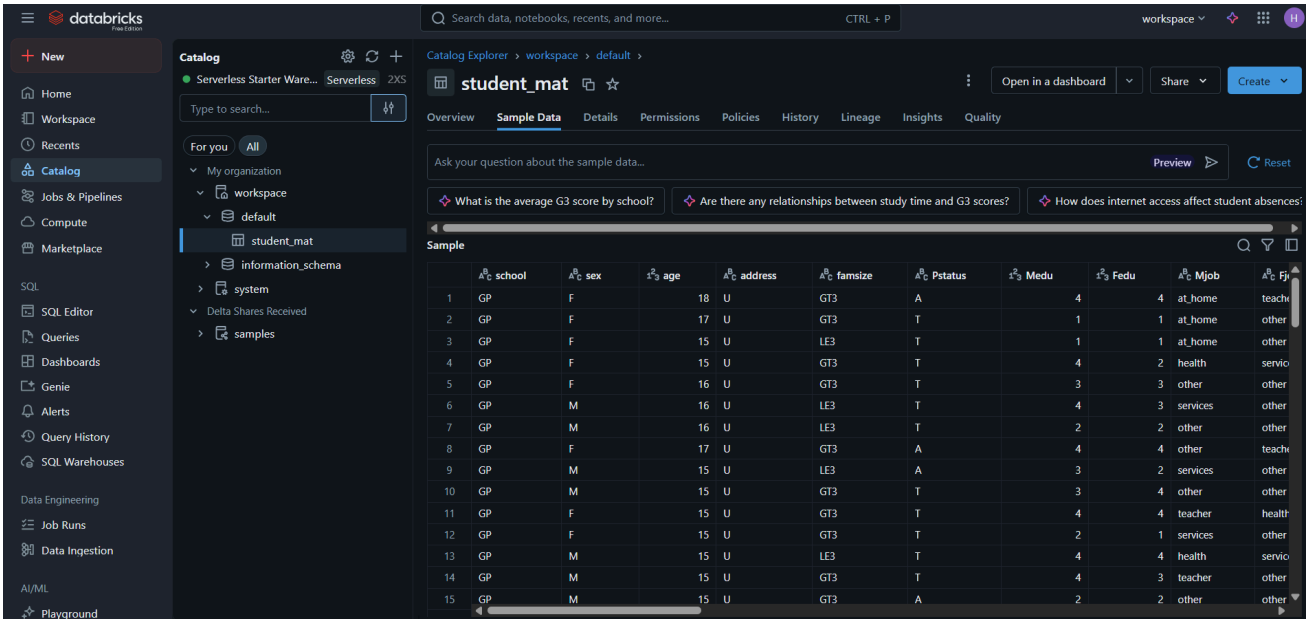
A etapa de coleta consistiu na obtenção do arquivo “student-mat.csv” da fonte original (UCI Repository).
Devido a instabilidades momentâneas na conexão direta com a URL da fonte, optou-se por realizar o upload manual do arquivo para o Databricks File System (DBFS), registrando-o no catálogo como “workspace.default.student_mat”

Código Fonte e Evidências:

- Nesta etapa, o dado bruto foi carregado e disponibilizado para leitura pelo Spark







4. Tratamento e Padronização (Camada Silver)

Nesta fase, o foco foi a limpeza e o enriquecimento dos dados brutos. As principais transformações realizadas foram:

- **Criação de Chave Primária (*Primary Key)*:** Como o dataset original não possuía um identificador único, foi gerada uma coluna “id_aluno” utilizando a função “monotonically_increasing_id()” do **PySpark**, essencial para o relacionamento entre tabelas no modelo dimensional.

- **Renomeação de Colunas:** Todas as 33 colunas originais (em inglês) foram mapeadas e renomeadas para o português (ex: “Medu” para “educacao_mae”, “famsup” para “suporte_familiar”), facilitando a interpretação por usuários de negócio.

- **Persistência Gerenciada:** A tabela tratada foi salva no formato **Delta Lake** como uma tabela gerenciada (“default.silver_student_performance”).

Código Fonte e Evidências:

https://github.com/HUGOGRIFOPERSONAL/HUGO_GRIFO_PUC_RIO_POS/blob/main/MVP_DataEng_2_Tratamento_Silver.ipynb

```python
# 2. Tratamento: Adicionar ID único e Renomear Colunas
# Como o dataset original não tem ID, criamos um para permitir o relacionamento futuro
df_silver = df_bronze.withColumn("id_aluno", monotonically_increasing_id()) \
    .withColumnRenamed("school", "escola") \
    .withColumnRenamed("sex", "sexo") \
    .withColumnRenamed("age", "idade") \
    .withColumnRenamed("address", "tipo_endereco") \
    .withColumnRenamed("famsize", "tamanho_familia") \
    .withColumnRenamed("Medu", "educacao_mae") \
    .withColumnRenamed("Fedu", "educacao_pai") \
    .withColumnRenamed("Mjob", "trabalho_mae") \
    .withColumnRenamed("Fjob", "trabalho_pai") \
    .withColumnRenamed("studytime", "tempo_estudo") \
    .withColumnRenamed("failures", "reprovacoes") \
    .withColumnRenamed("schoolsup", "suporte_escolar") \
    .withColumnRenamed("famsup", "suporte_familiar") \
    .withColumnRenamed("paid", "aulas_extras_pagas") \
    .withColumnRenamed("activities", "atividades_extracurriculares") \
    .withColumnRenamed("internet", "possui_internet") \
    .withColumnRenamed("Dalc", "alcool_dia_util") \
    .withColumnRenamed("Walc", "alcool_fim_semana") \
    .withColumnRenamed("health", "saude") \
    .withColumnRenamed("absences", "faltas") \
    .withColumnRenamed("G1", "nota_g1") \
    .withColumnRenamed("G2", "nota_g2") \
    .withColumnRenamed("G3", "nota_final")

# 3. Removendo Duplicadas por linhas identicas
```

```
|-- G3: long (nullable = true)

Tabela Silver salva com sucesso: default.silver_student_performance
```

Table +

	escola	sexo	idade	tipo_endereco	tamanho_familia	Pstatus	educacao_mae	educacao_pai
1	GP	F	18	R	GT3	T	4	4
2	GP	F	17	U	LE3	T	4	2
3	GP	F	18	U	GT3	T	2	1
4	GP	F	17	U	GT3	T	4	3
5								

5 rows | 11.44s runtime Refreshed now

5. Modelagem Dimensional (Camada Gold)

Para viabilizar as análises, os dados da camada Silver foram reestruturados em um **Esquema Estrela (*Star Schema*)**. O "tabelão" único foi decomposto em quatro tabelas especializadas:

- **“default.gold_fato_desempenho”:** Contém as métricas quantitativas, como notas (G1, G2, G3) e número de reprovações.
- **“default.gold_dim_aluno”:** Contém atributos demográficos (idade, sexo, escola, endereço).
- **“default.gold_dim_socioeconomico”:** Contém dados sobre o contexto familiar (educação e trabalho dos pais, internet).

- **“default.gold_dim_habitos”:** Contém dados comportamentais (consumo de álcool, tempo de estudo, faltas).

Essa modelagem facilita o cruzamento de dados e melhora a performance de ferramentas de BI e consultas SQL.

Código Fonte e Evidências:

https://github.com/HUGOGRIFOPERSONAL/HUGO_GRIFO_PUC_RIO_POS/blob/main/MVP_DataEng_3_Modelagem_Gold.ipynb

```
# 1. Leitura da Tabela Silver
df_silver = spark.read.table("workspace.default.silver_student_performance")

# 2. Criação das Dimensões e Fato

# Dimensão Aluno
df_dim_aluno = df_silver.select(
    "id_aluno", "escola", "sexo", "idade", "tipo_endereco", "tamanho_familia", "razao_escolha_escola", "guardiao"
)

# Dimensão Socioeconômica
df_dim_socioeconomico = df_silver.select(
    "id_aluno", "educacao_mae", "educacao_pai", "trabalho_mae", "trabalho_pai", "possui_internet", "suporte_familiar", "status_coabitacao_pais"
)

# Dimensão Hábitos
df_dim_habitos = df_silver.select(
    "id_aluno", "tempo_estudo", "alcool_dia_util", "alcool_fim_semana", "saude", "atividades_extracurriculares", "faltas", "tempo_livre",
    "sair_com_amigos"
)

# Tabela Fato
df_fato_desempenho = df_silver.select(
    "id_aluno", "nota_g1", "nota_g2", "nota_final", "reprovacoes"
)

# 3. Carga (Salvar como Tabelas Gold Gerenciadas)
df_dim_aluno.write.format("delta").mode("overwrite").saveAsTable("default.gold_dim_aluno")
```

```
df_fato_desempenho.write.format("delta").mode("overwrite").saveAsTable("default.gold_fato_desempenho")

print("Tabelas Gold criadas com sucesso no esquema 'default'!")
```

See performance (4)

```
df_silver: pyspark.sql.connect.dataframe.DataFrame = [escola: string, sexo: string ... 32 more fields]
df_dim_aluno: pyspark.sql.connect.dataframe.DataFrame = [id_aluno: long, escola: string ... 6 more fields]
df_dim_socioeconomico: pyspark.sql.connect.dataframe.DataFrame = [id_aluno: long, educacao_mae: long ... 6 more fields]
df_dim_habitos: pyspark.sql.connect.dataframe.DataFrame = [id_aluno: long, tempo_estudo: long ... 7 more fields]
df_fato_desempenho: pyspark.sql.connect.dataframe.DataFrame = [id_aluno: long, nota_g1: long ... 3 more fields]
Tabelas Gold criadas com sucesso no esquema 'default'!
```

6. Catálogo de Dados

Tabela Fato: “default.gold_fato_desempenho”

Descrição: Contém as métricas quantitativas de desempenho e histórico escolar do aluno.

Coluna	Tipo de Dado	Descrição do Atributo
id_aluno	Long	Chave Estrangeira única para identificação do aluno.
nota_g1	Integer	Nota do 1º período (escala de 0 a 20).
nota_g2	Integer	Nota do 2º período (escala de 0 a 20).
nota_final	Integer	Nota final do ano (G3, escala de 0 a 20). Target principal da análise.
reprovacoes	Integer	Número de classes reprovadas anteriormente (n se 1<=n<3, senão 4).

Tabela Dimensão: default.gold_dim_aluno

Descrição: Dados demográficos e pessoais do estudante.

Coluna	Tipo de Dado	Descrição do Atributo
id_aluno	Long	Chave Primária única do aluno.

escola	String	Escola do aluno ('GP' - Gabriel Pereira ou 'MS' - Mousinho da Silveira).
sexo	String	Sexo do aluno ('F' - Feminino ou 'M' - Masculino).
idade	Integer	Idade do aluno (de 15 a 22 anos).
tipo_endereco	String	Tipo de endereço ('U' - Urbano ou 'R' - Rural).
tamanho_familia	String	Tamanho da família ('LE3' - menor ou igual a 3 ou 'GT3' - maior que 3).
razao_escolha_escola	String	Razão para escolher a escola (perto de casa, reputação, curso ou outro).
guardiao	String	Guardião legal do aluno (mãe, pai ou outro).

Tabela Dimensão: “default.gold_dim_socioeconomico”

Descrição: Contexto familiar e socioeconômico.

Coluna	**Tipo de Dado**	**Descrição do Atributo**
id_aluno	Long	Chave Primária única do aluno.

educacao_mae	Integer	Escolaridade da mãe (0: Nenhuma, 1: Primário, 2: 5º-9º ano, 3: Médio, 4: Superior).
educacao_pai	Integer	Escolaridade do pai (0: Nenhuma, 1: Primário, 2: 5º-9º ano, 3: Médio, 4: Superior).
trabalho_mae	String	Tipo de trabalho da mãe (ex: 'teacher', 'health', 'services', 'at_home', 'other').
trabalho_pai	String	Tipo de trabalho do pai (ex: 'teacher', 'health', 'services', 'at_home', 'other').
possui_internet	String	Possui acesso à internet em casa ('yes' ou 'no').
suporte_familiar	String	Recebe suporte educacional da família ('yes' ou 'no').
status_coabitacao_pais	String	Status de coabitação dos pais ('T' - moram juntos ou 'A' - separados).

Tabela Dimensão: “default.gold_dim_habitos”

Descrição: Dados comportamentais, rotina e saúde.

Coluna	Tipo de Dado	Descrição do Atributo
id_aluno	Long	Chave Primária única do aluno.
tempo_estudo	Integer	Tempo semanal de estudo (1: <2h, 2: 2-5h, 3: 5-10h, 4: >10h).
alcool_dia_util	Integer	Consumo de álcool em dias úteis (escala de 1 - muito baixo a 5 - muito alto).
alcool_fim_semana	Integer	Consumo de álcool no fim de semana (escala de 1 - muito baixo a 5 - muito alto).
saude	Integer	Estado de saúde atual (escala de 1 - muito ruim a 5 - muito bom).
faltas	Integer	Quantidade de faltas escolares (de 0 a 93).
atividades_extracurriculares	String	Participa de atividades extracurriculares ('yes' ou 'no').
tempo_livre	Integer	Tempo livre após a escola (escala de 1 - muito baixo a 5 - muito alto).

sair_com_amigos	Integer	Frequência de sair com amigos (escala de 1 - muito baixo a 5 - muito alto).

7. Análise de Dados e Solução do Problema

7.1. Qualidade dos Dados: Verificou-se a existência de valores nulos ou notas fora do intervalo padrão (0-20). A análise confirmou a integridade do dataset, com total consistência nos registros de notas finais.

7.2. Solução do Problema:

- **Impacto do Álcool:** Observou-se uma tendência onde o alto consumo de álcool aos finais de semana está correlacionado com uma leve queda na média das notas finais.
- **Escolaridade Materna:** Identificou-se uma correlação positiva clara: alunos cujas mães possuem ensino superior tendem a obter médias finais superiores em comparação àqueles cujas mães não possuem instrução formal.
- **Tempo de Estudo:** Confirmou-se que o tempo dedicado aos estudos é um fator determinante, com as maiores notas concentradas no grupo que estuda mais de 10 horas semanais.

Código Fonte e Evidências:

https://github.com/HUGOGRIFOPERSONAL/HUGO_GRIFO_PUC_RIO_POS/blob/main/MVP_DataEng_5_SQL_Gold.ipynb

Just now (2s) 2 SQL

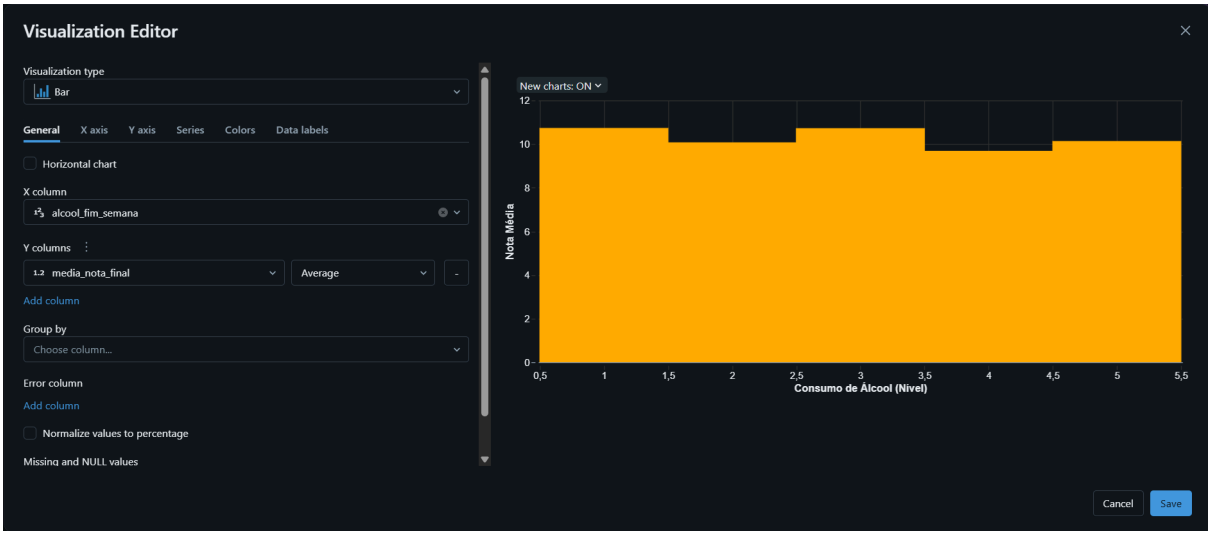
```
%sql
SELECT
    h.alcool_fim_semana,
    ROUND(AVG(f.nota_final), 2) as media_nota_final,
    COUNT(f.id_aluno) as qtd_alunos
FROM default.gold_fato_desempenho f
JOIN default.gold_dim_habitos h ON f.id_aluno = h.id_aluno
GROUP BY h.alcool_fim_semana
ORDER BY h.alcool_fim_semana ASC
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [alcool_fim_semana: long, media_nota_final: double ... 1 more field]

Table +

	alcool_fim_semana	media_nota_final	qtd_alunos
1	1	10.74	151
2	2	10.08	85
3	3	10.73	80
4	4	9.69	51
5	5	10.14	28



Just now (2s) 3 SQL

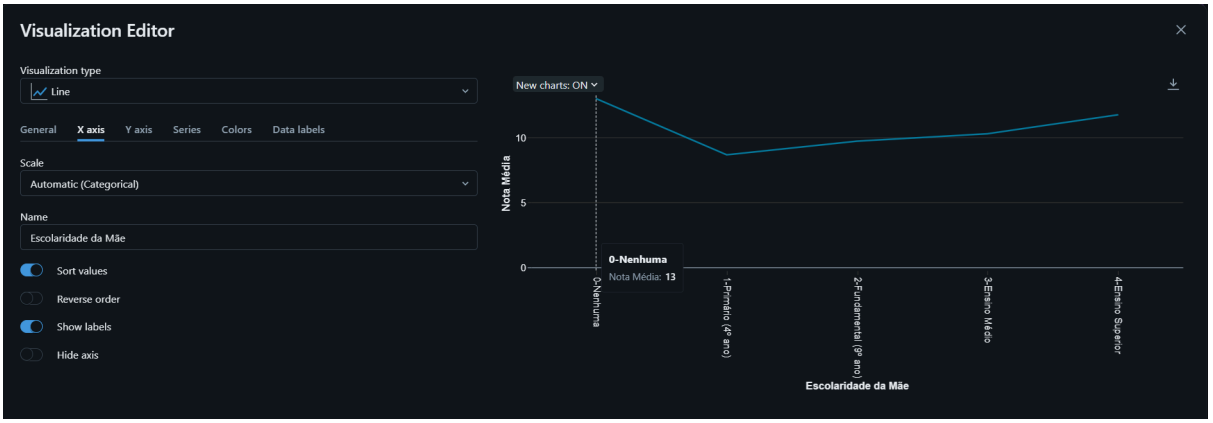
```
%sql
SELECT
    CASE
        WHEN s.educacao_mae = 0 THEN '0-Nenhuma'
        WHEN s.educacao_mae = 1 THEN '1-Primário (4º ano)'
        WHEN s.educacao_mae = 2 THEN '2-Fundamental (9º ano)'
        WHEN s.educacao_mae = 3 THEN '3-Ensino Médio'
        WHEN s.educacao_mae = 4 THEN '4-Ensino Superior'
    END as escolaridade_mae,
    ROUND(AVG(f.nota_final), 2) as media_nota_final
FROM default.gold_fato_desempenho f
JOIN default.gold_dim_socioeconomico s ON f.id_aluno = s.id_aluno
GROUP BY s.educacao_mae
ORDER BY s.educacao_mae ASC
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [escolaridade_mae: string, media_nota_final: double]

Table +

	escolaridade_mae	media_nota_final
1	0-Nenhuma	13
2	1-Primário (4º ano)	8.68
3	2-Fundamental (9º ano)	9.73
4	3-Ensino Médio	10.3
5	4-Ensino Superior	11.76



Just now (1s) 4 SQL

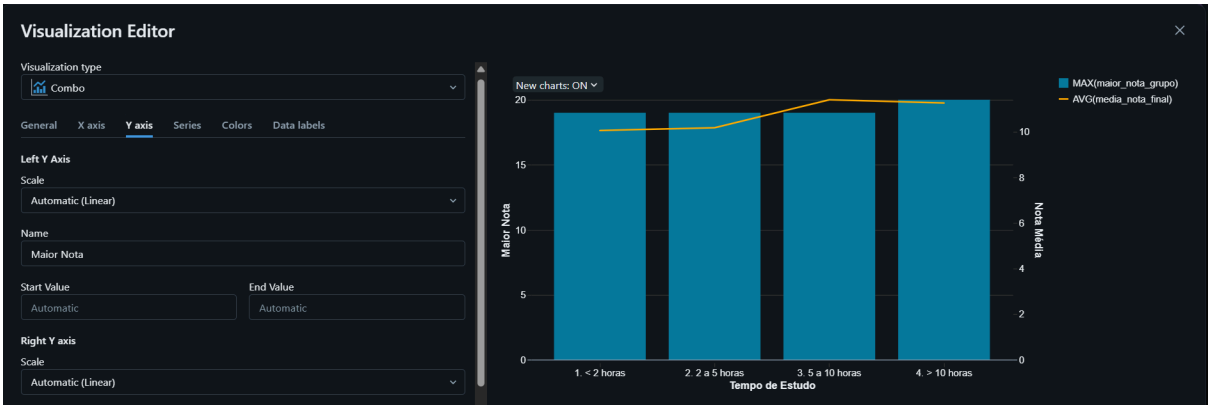
```
%sql
SELECT
    CASE
        WHEN h.tempo_estudo = 1 THEN '1. < 2 horas'
        WHEN h.tempo_estudo = 2 THEN '2. 2 a 5 horas'
        WHEN h.tempo_estudo = 3 THEN '3. 5 a 10 horas'
        WHEN h.tempo_estudo = 4 THEN '4. > 10 horas'
    END as tempo_estudo_semanal,
    ROUND(AVG(f.nota_final), 2) as media_nota_final,
    MAX(f.nota_final) as maior_nota_grupo
FROM default.gold_fato_desempenho f
JOIN default.gold_dim_habitos h ON f.id_aluno = h.id_aluno
GROUP BY h.tempo_estudo
ORDER BY h.tempo_estudo ASC
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [tempo_estudo_semanal: string, media_nota_final: double ... 1 more field]

Table +

	tempo_estudo_semanal	media_nota_final	maior_nota_grupo
1	1. < 2 horas	10.05	19
2	2. 2 a 5 horas	10.17	19
3	3. 5 a 10 horas	11.4	19
4	4. > 10 horas	11.26	20



Just now (2s) 5 SQL

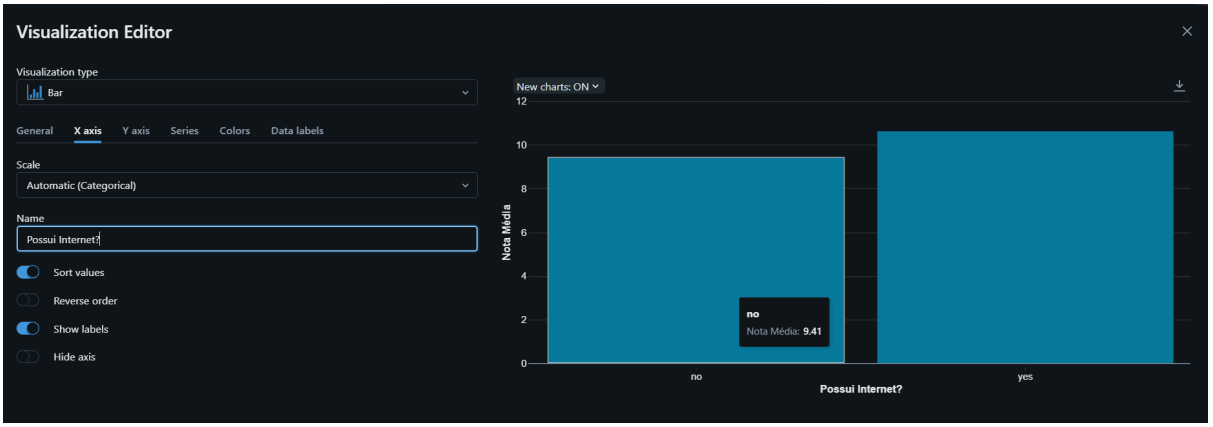
```
%sql
SELECT
    s.possui_internet,
    ROUND(AVG(f.nota_final), 2) as media_nota_final
FROM default.gold_fato_desempenho f
JOIN default.gold_dim_socioeconomico s ON f.id_aluno = s.id_aluno
GROUP BY s.possui_internet
```

See performance (1) Optimize

_sqldf: pyspark.sql.connect.dataframe.DataFrame = [possui_internet: string, media_nota_final: double]

Table +

	possui_internet	media_nota_final
1	no	9.41
2	yes	10.62



8. Conclusão e Autoavaliação

Este projeto (MVP) me permitiu aplicar os conceitos fundamentais da Engenharia de Dados moderna. A implementação de uma arquitetura em camadas (*Bronze*, *Silver* e *Gold*) no Databricks provou-se eficaz para organizar o fluxo de tratamento da informação.

Durante o desenvolvimento, enfrentei desafios técnicos, especificamente relacionados ao schema enforcement (imposição de esquema) das tabelas Delta, que bloqueavam a sobrescrita de tabelas quando novas colunas eram adicionadas. A solução encontrada foi a utilização da opção “overwriteSchema”, o que garantiu a flexibilidade necessária durante a fase de desenvolvimento.

Agradeço à Pontifícia Universidade Católica do Rio de Janeiro pelas aulas ministradas, pelo conteúdo disponibilizado e pela oportunidade de realizar o projeto.