

SISTEMA DE CONTROLE DE FINANÇAS PESSOAIS

MEWLHÃO



AUTORA: LAURA ANITA COSTA

Ferramentas / Tecnologias:

Excel, GoogleColab, VSCode,
Python, SQLite, DBeaver,
PowerShell, Docker, Metabase

Morrinhos/GO

2025

SUMÁRIO

1. INTRODUÇÃO.....	3
2. OBJETIVOS DO PROJETO.....	3
3. CRIAÇÃO DOS DADOS NO EXCEL E CONVERSÃO PARA CSV.....	3
4. TRATAMENTO DOS DADOS NO GOOGLE COLAB.....	4
5. CRIAÇÃO DO BANCO DE DADOS (PYTHON COM SQLITE).....	6
6. VALIDAÇÃO DOS DADOS NO VS CODE.....	7
7. IMPORTAÇÃO DO BANCO DE DADOS NO DBEAVER.....	7
8. TRATAMENTO DE DADOS COM SQLITE NO DBEAVER.....	10
9. VALIDAÇÃO FINAL DO BANCO DE DADOS NO VS CODE.....	12
10. CRIAÇÃO DO CONTAINER METABASE NO DOCKER USANDO O POWERSHELL.....	12
11. CONECTANDO O BANCO DE DADOS SQLITE NO METABASE.....	13
12. CRIANDO DASHBOARD NO METABASE.....	16
13. CONSIDERAÇÕES FINAIS.....	22
14. REFERÊNCIAS.....	23

1. INTRODUÇÃO

O projeto **Mewlhão** foi desenvolvido como uma forma prática e divertida de aplicar conhecimentos em engenharia e análise de dados, utilizando uma narrativa lúdica com o mascote Mewlhão e uma tabela de despesas, contendo todos os gastos separados por categorias, subcategorias.

A proposta engloba desde a construção de uma base de dados fictícia até a visualização final em dashboards interativos, passando por etapas fundamentais do ciclo de dados.

2. OBJETIVOS DO PROJETO

- Desenvolver habilidades em todas as etapas de um pipeline de dados
- Aplicar conceitos de ETL e análise de dados em um projeto real
- Consolidar o uso de ferramentas como Python, SQLite, DBeaver, Docker e Metabase
- Criar um projeto completo, reproduzível e didático
- Estimular o aprendizado de forma leve e criativa

3. CRIAÇÃO DOS DADOS NO EXCEL E CONVERSÃO PARA CSV

Nesta etapa, foi criada uma tabela do zero manualmente no Excel com os dados iniciais desejados, contendo alguns erros propositais, como células vazias.

Foram adicionadas as seguintes colunas: DATA, DESCRIÇÃO, CATEGORIA, SUBCATEGORIA e VALOR.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	DATA	DESCRIÇÃO	CATEGORIA	SUBCATEGORIA	VALOR												
71	2025-11-09	Roupas Nayara Marques	Vestuário	Roupas de Academia	48.33												
72	2025-11-09	Biopharma	Saúde	Medicamentos	21.35												
73	2025-11-09	MSN Indústria e Comércio de Roupas	Vestuário	Roupas	52.58												
75	2025-11-09	Jeff Cândido	Saúde	Plano Academia	10.47												
76	2025-11-08	Roupas Mari Amaral	Vestuário	Roupas	17.64												
77	2025-11-08	Roupas Micaelly Mih Store	Vestuário	Roupas	26.80												
78	2025-11-08	Pagarme Pagamentos SA			57.73												
79	2025-11-08	Drogaria Norreste	Saúde	Medicamentos	30.49												
80	2025-11-05	Livros Amazon	Lazer	Leitura	50.43												
81	2025-11-03	RodBurn	Saúde	Suplemento	42.29												
82	2025-11-03	Treino RodBurn	Saúde	Personal	9.54												
83	2025-11-01	Shopee Brendaturinipa Papelaria	Lazer	Papelaria	13.94												
84	2025-11-01	Amazon Livros	Lazer	Leitura	36.43												
85	2025-11-01	Amanda Cílios	Beleza	Cílios	9.64												
86	2025-11-01	Amazon Livros	Lazer	Leitura	14.36												
87	2025-11-01	Pedro Igor Amorim			10.93												
88	2025-11-01	Shopee Tonsderosa	Lazer	Condicionador Colorido Cabelo	16.42												
89	2025-11-01	Drogaria Ultra Popular	Saúde	Medicamentos	4.50												
90	2025-11-01	Leonardo Paixão	Pets	Akira	29.18												
91	2025-11-01	Amazon Livros	Lazer	Leitura	12.38												
93	2025-11-01	Roupas Jordana da Castro	Vestuário	Roupas	14.17												

Posteriormente essa tabela foi exportada com o formato .csv sendo salva com separador ‘;’ de forma proposital para futuro tratamento.

C:\Users\Laurinha\Desktop\Projeto_Financas_Pessoais\despesas.CSV - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Ferramentas Macro Executar Plugins Janela ?

despesas.CSV

```

1 DATA;DESCRICAÇÃO;CATEGORIA;SUBCATEGORIA;VALOR
2 2026-05-20;Kindle Unlimited Amazon;Lazer;Leitura;19.90
3 2026-05-10;Mensalidade Faculdade Estácio;Estudo;Faculdade;148.24
4 2026-05-10;iFood Club;Lazer;Assinatura;5.95
5 2026-05-10;Jeff Cândido;Saúde;Plano Academia;60.00
6 2026-04-20;Kindle Unlimited Amazon;Lazer;Leitura;19.90
7 2026-04-10;Mensalidade Faculdade Estácio;Estudo;Faculdade;148.24
8 2026-04-10;iFood Club;Lazer;Assinatura;5.95
9 2026-04-10;Jeff Cândido;Saúde;Plano Academia;60.00
10 2026-04-09;Roupas Micaelly Mih Store;Vestuário;Roupas;13.03
11 2026-04-09;MSN Indústria e Comércio de Roupas;Vestuário;Roupas;52.58
12 2026-03-20;Kindle Unlimited Amazon;Lazer;Leitura;19.90
13 2026-03-10;Mensalidade Faculdade Estácio;Estudo;Faculdade;148.24
14 2026-03-10;iFood Club;Lazer;Assinatura;5.95
15 2026-03-10;Jeff Cândido;Saúde;Plano Academia;60.00
16 2026-03-09;Roupas Micaelly Mih Store;Vestuário;Roupas;13.03
17 2026-03-09;MSN Indústria e Comércio de Roupas;Vestuário;Roupas;52.58
18 2026-03-08;Roupas Micaelly Mih Store;Vestuário;Roupas;26.80
19 2026-03-01;Amazon Livros;Lazer;Leitura;36.43
20 2026-03-01;Shopee Lojinhamaia;Lazer;Papelaria;26.56
21 2026-02-20;Kindle Unlimited Amazon;Lazer;Leitura;19.90
22 2026-02-10;Mensalidade Faculdade Parácia;Estudo;Faculdade;148.24

```

Normal text file length: 27.721 lines: 518 Ln: 1 Col: 44 Pos: 44 Windows (CR LF) ANSI INS

4. TRATAMENTO DOS DADOS NO GOOGLE COLAB

Nesta etapa, o arquivo original despesas.csv foi carregado no Google Colab para ser tratado e padronizado. A codificação ISO-8859-1 foi utilizada devido ao uso de caracteres especiais na base.

As ações executadas foram:

- Upload do arquivo CSV via Google Colab - [Projeto_Financas_Pessoais.ipynb](#);
- Leitura do arquivo utilizando a biblioteca pandas;

- Preenchimento de valores nulos nas colunas CATEGORIA e SUBCATEGORIA com o valor 'Desconhecido';
- Correção manual de um erro de digitação na coluna SUBCATEGORIA, alterando 'Ciliros' para 'Cílios';
- Conversão da coluna DATA para o tipo datetime, com tratamento de erros;
- Exportação da base tratada para um novo arquivo CSV: despesas_tratadas.csv;
- Download do arquivo final para uso posterior no banco de dados.

Essas etapas garantiram que os dados estivessem limpos, padronizados e prontos para importação no SQLite.

```

[ ] import os
os.listdir('/content')

[ ].config', 'despesas.CSV', 'sample_data']

[ ] from google.colab import files

# Realize o upload do arquivo
uploaded = files.upload()

Nenhum ficheiro selecionado Upload widget is only available when the cell been executed in the current browser session. Please rerun this cell to enable.
Saving despesas.CSV to despesas.CSV

[ ] import pandas as pd

# Ler o arquivo CSV com codificação 'ISO-8859-1'
df = pd.read_csv('/content/despesas.CSV', sep=';', encoding='ISO-8859-1')

# Visualizar as primeiras linhas
df.head()

```

	DATA	DESCRICA	CATEGORIA	SUBCATEGORIA	VALOR
0	2026-05-20	Kindle Unlimited Amazon	Lazer	Leitura	19.90
1	2026-05-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148.24
2	2026-05-10	iFood Club	Lazer	Assinatura	5.95
3	2026-05-10	Jeff Cândido	Saúde	Piano Academia	60.00
4	2026-04-20	Kindle Unlimited Amazon	Lazer	Leitura	19.90

```

[ ] # Verificar valores nulos
df.isnull().sum()

```

	DATA	DESCRICA	CATEGORIA	SUBCATEGORIA	VALOR
	0	0	0	0	0

```

[ ] # Preencher valores nulos nas colunas "CATEGORIA" e "SUBCATEGORIA" com "Desconhecido"
df['CATEGORIA'] = df['CATEGORIA'].fillna('Desconhecido')
df['SUBCATEGORIA'] = df['SUBCATEGORIA'].fillna('Desconhecido')

[ ] # Verificar valores nulos
df.isnull().sum()

```

	DATA	DESCRICA	CATEGORIA	SUBCATEGORIA	VALOR
	0	0	0	0	0

```

[ ] # Verificar categorias únicas
print(df['CATEGORIA'].unique())
print(df['SUBCATEGORIA'].unique())

[ ] # Corrigir um erro de digitação (exemplo: 'transporte' para 'Transporte')
df['SUBCATEGORIA'] = df['SUBCATEGORIA'].replace('Ciliros', 'Cílios')

[ ] # Verificar categorias únicas
print(df['CATEGORIA'].unique())
print(df['SUBCATEGORIA'].unique())

[ ] # Verificar categorias únicas
print(df['CATEGORIA'].unique())
print(df['SUBCATEGORIA'].unique())

```

```

[ ] # Converter a coluna 'DATA' para o formato datetime
df['DATA'] = pd.to_datetime(df['DATA'], format='%d-%m-%Y', errors='coerce')

# Verificar as primeiras linhas novamente
print(df.head())

[ ] DATA DESCRICAO CATEGORIA SUBCATEGORIA VALOR
0 2026-05-20 Kindle Unlimited Amazon Lazer Leitura 19.90
1 2026-05-10 Mensalidade Faculdade Estácio Estudo Faculdade 148.24
2 2026-05-10 iFood Club Lazer Assinatura 5.95
3 2026-05-10 Jeff Cândido Saúde Plano Academia 60.00
4 2026-04-20 Kindle Unlimited Amazon Lazer Leitura 19.90

[ ] # Salvar o dataframe tratado em um novo arquivo CSV
df.to_csv('/content/despesas_tratadas.csv', index=False, sep=';')

[ ] from google.colab import files
# Fazer o download do arquivo
files.download('/content/despesas_tratadas.csv')

```

5. CRIAÇÃO DO BANCO DE DADOS (PYTHON COM SQLITE)

Após o tratamento dos dados no Google Colab, foi criado um banco de dados local no formato SQLite utilizando Python no VSCode.

O objetivo dessa etapa foi consolidar os dados em uma estrutura relacional simples e eficiente para facilitar futuras consultas e visualizações.

As etapas foram:

- Leitura do arquivo ‘despesas_tratadas.csv’ com o ‘pandas’, utilizando o separador correto (;);
- Conversão da coluna ‘DATA’ para o formato ‘datetime’, garantindo a interpretação correta das datas;
- Conversão da coluna ‘VALOR’ para o tipo ‘float’, assegurando o correto tratamento dos valores numéricos;
- Criação de um banco de dados SQLite local chamado despesas.db;
- Inserção de todos os dados tratados no banco de dados por meio do método ‘to_sql()’ do ‘pandas’, criando a tabela despesas.

```

criando_db.py > ...
import sqlite3
import pandas as pd

# Lendo o CSV tratado adaptando o separador
df = pd.read_csv('despesas_tratadas.csv', sep=';')

# Garantindo que a coluna 'DATA' esteja com formato datetime
df['DATA'] = pd.to_datetime(df['DATA'], errors='coerce')

# Converter valores numéricos para float
df['VALOR'] = df['VALOR'].astype(float)

# Criando o banco de dados
conn = sqlite3.connect('despesas.db')

# Criando a tabela 'despesas' e inserindo seus dados
df.to_sql('despesas', conn, if_exists='replace', index=False)

# Fechando a conexão
conn.commit()
conn.close()

print("Dados importados com sucesso!")

```

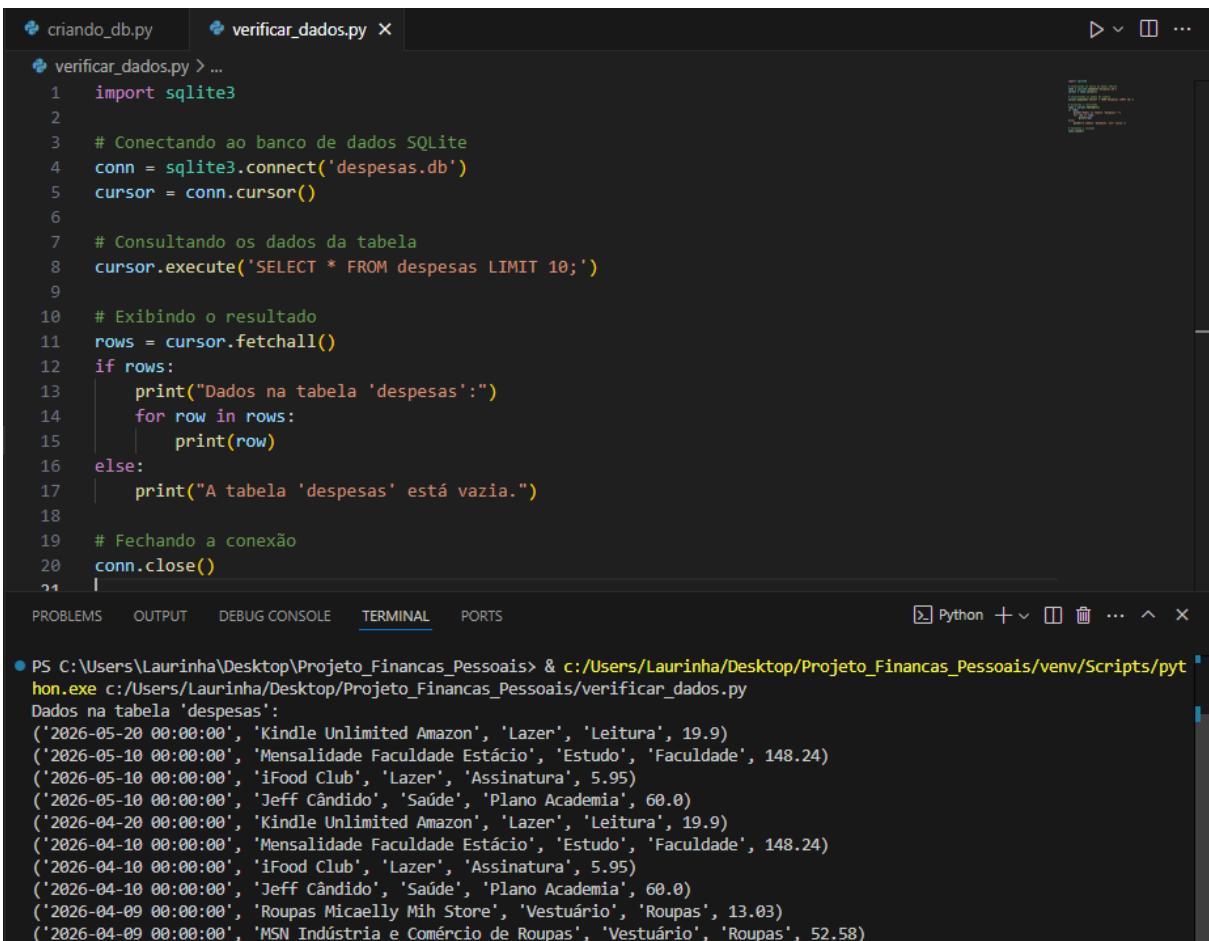
PS C:\Users\Laurinha\Desktop\Projeto_Financas_Pessoais> & c:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais/criando_db.py
hon.exe c:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais/criando_db.py
Dados importados com sucesso!

Esse processo resultou em um banco SQLite com os dados organizados e prontos para análises posteriores.

6. VALIDAÇÃO DOS DADOS NO VS CODE

Após a criação do banco de dados, foi realizada a validação dos dados utilizando o VSCode. O objetivo dessa etapa foi garantir que os dados foram corretamente inseridos no banco SQLite e que a estrutura da tabela estava conforme o esperado.

A validação consistiu em conectar-se ao banco de dados despesas.db, realizar uma consulta simples para exibir as primeiras 10 linhas da tabela despesas e verificar se os dados estavam disponíveis e organizados corretamente.



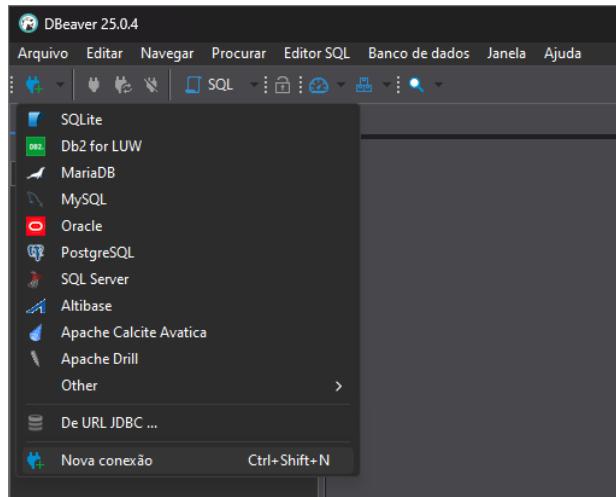
```
criando_db.py  verificar_dados.py ...  
verificar_dados.py > ...  
1 import sqlite3  
2  
3 # Conectando ao banco de dados SQLite  
4 conn = sqlite3.connect('despesas.db')  
5 cursor = conn.cursor()  
6  
7 # Consultando os dados da tabela  
8 cursor.execute('SELECT * FROM despesas LIMIT 10;')  
9  
10 # Exibindo o resultado  
11 rows = cursor.fetchall()  
12 if rows:  
13     print("Dados na tabela 'despesas':")  
14     for row in rows:  
15         print(row)  
16 else:  
17     print("A tabela 'despesas' está vazia.")  
18  
19 # Fechando a conexão  
20 conn.close()  
21 |  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python ...  
PS C:\Users\Laurinha\Desktop\Projeto_Financas_Pessoais> & c:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais/venv/Scripts/python.exe c:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais/verificar_dados.py  
Dados na tabela 'despesas':  
('2026-05-20 00:00:00', 'Kindle Unlimited Amazon', 'Lazer', 'Leitura', 19.9)  
('2026-05-10 00:00:00', 'Mensalidade Faculdade Estácio', 'Estudo', 'Faculdade', 148.24)  
('2026-05-10 00:00:00', 'iFood Club', 'Lazer', 'Assinatura', 5.95)  
('2026-05-10 00:00:00', 'Jeff Cândido', 'Saúde', 'Plano Academia', 60.0)  
('2026-04-20 00:00:00', 'Kindle Unlimited Amazon', 'Lazer', 'Leitura', 19.9)  
('2026-04-10 00:00:00', 'Mensalidade Faculdade Estácio', 'Estudo', 'Faculdade', 148.24)  
('2026-04-10 00:00:00', 'iFood Club', 'Lazer', 'Assinatura', 5.95)  
('2026-04-10 00:00:00', 'Jeff Cândido', 'Saúde', 'Plano Academia', 60.0)  
('2026-04-09 00:00:00', 'Roupas Micaelly Mih Store', 'Vestuário', 'Roupas', 13.03)  
('2026-04-09 00:00:00', 'MSN Indústria e Comércio de Roupas', 'Vestuário', 'Roupas', 52.58)
```

7. IMPORTAÇÃO DO BANCO DE DADOS NO DBEAVER

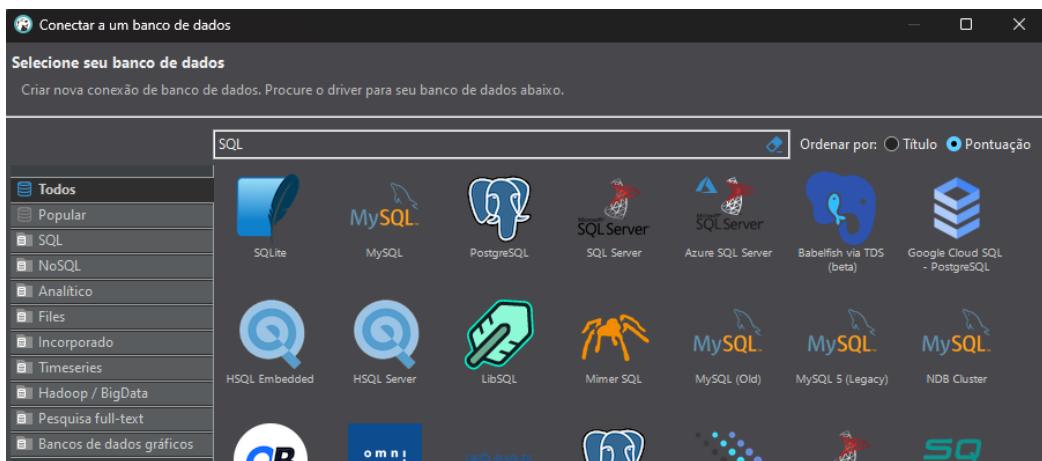
Com o banco de dados despesas.db criado e validado, o próximo passo foi importá-lo para o DBBeaver para facilitar a exploração visual dos dados e possíveis edições via interface gráfica.

Passo a passo para importar o banco SQLite no DBeaver:

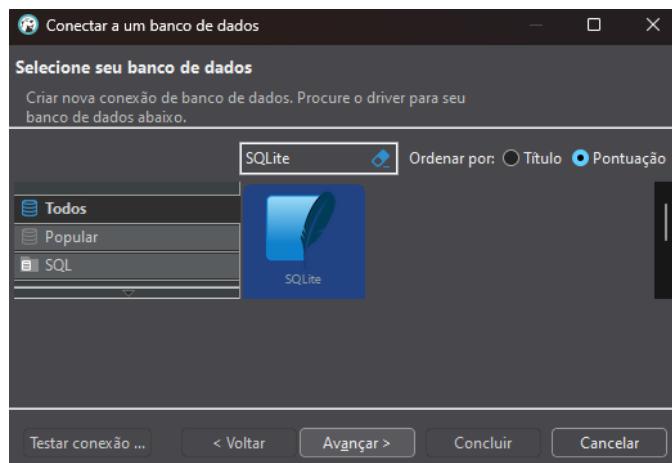
- Abrir o DBeaver e clicar no botão "Nova Conexão" (ícone com símbolo de plug ou Arquivo > Nova Conexão).



- Na lista de bancos de dados, procurar e selecionar SQLite.



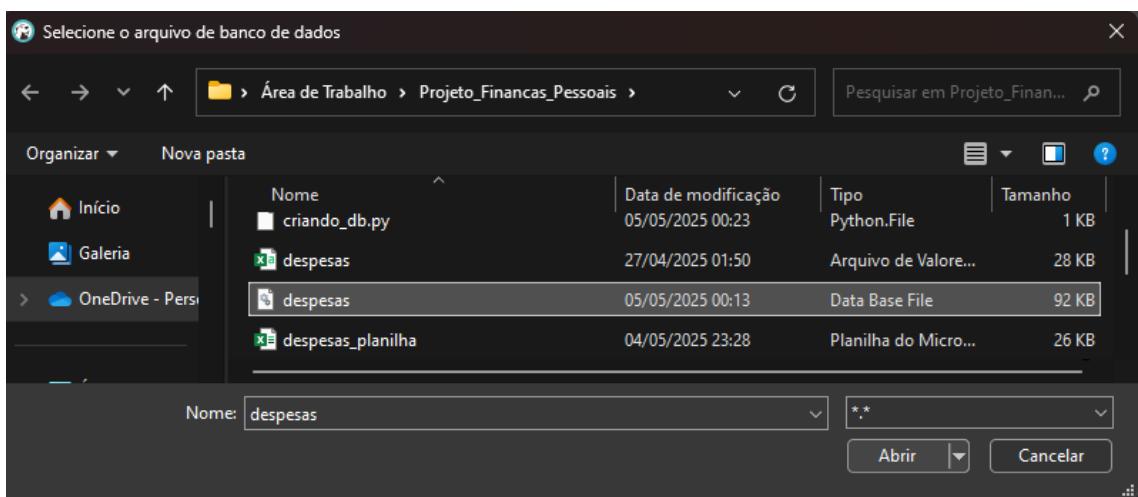
- Clicar em "Avançar".



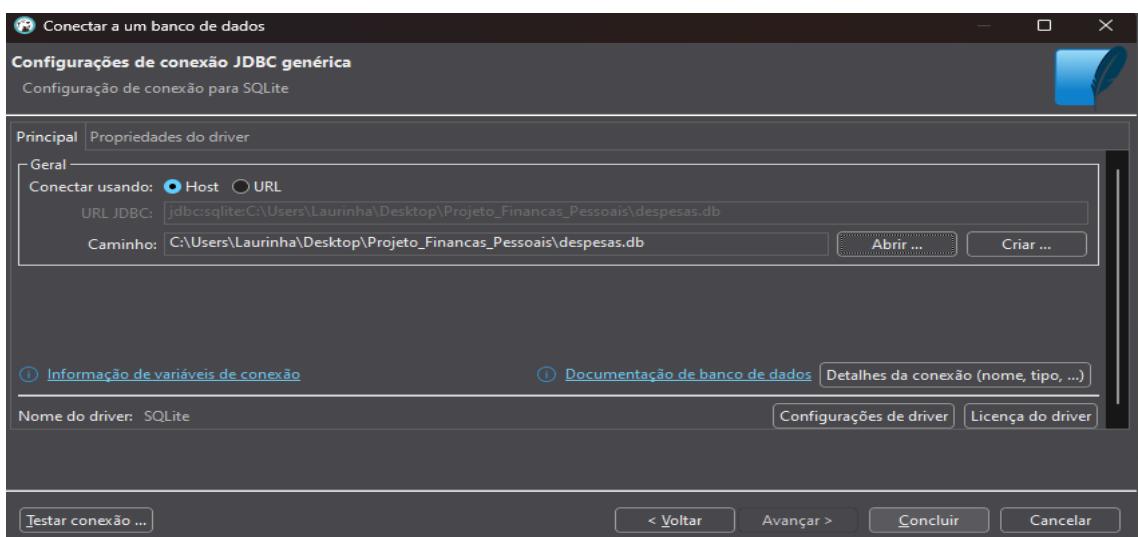
- Na tela de configuração, clicar no botão "Abrir" ao lado de "Caminho".



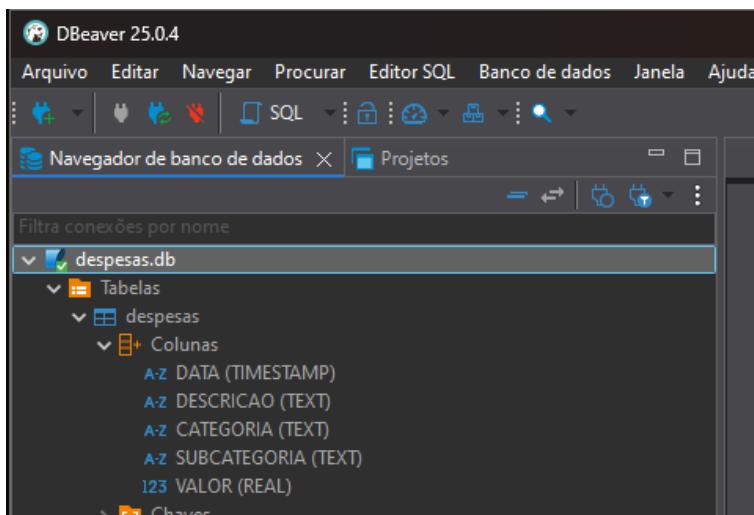
- Localizar e selecionar o arquivo despesas.db criado anteriormente no projeto.



- Finalizar a conexão clicando em "Concluir".



A partir desse momento, o banco estará acessível no painel esquerdo do DBeaver, permitindo a execução de queries SQL, visualização dos dados em tabelas, estrutura, tipos de dados e muito mais.



8. TRATAMENTO DE DADOS COM SQLITE NO DBEAVER

Com os dados organizados no banco SQLite criado via VSCode, a etapa seguinte foi utilizar o DBeaver para executar consultas SQL e realizar ajustes finos diretamente na tabela.

As ações executadas incluíram:

- Validação e inspeção das categorias e subcategorias existentes;
- Correções manuais em registros específicos, ajustando categorias, subcategorias e descrições;
- Adição da coluna "RECORRENIA" para indicar se os gastos são mensais, pontuais ou anuais;
- Preenchimento dessa nova coluna com base em regras específicas;
- Inclusão de novos registros de forma manual para complementar o conjunto de dados.
- Conversão dos valores da coluna DATA para o formato apenas de data (YYYY-MM-DD), removendo a informação de horário (00:00:00) que havia sido mantida como parte do tipo Timestamp.

Essa etapa permitiu enriquecer e refinar ainda mais a base de despesas antes da análise no Metabase.

DBeaver 25.0.4 - <despesas.db> Script-1

Arquivo Editar Navegar Procurar Editor SQL Banco de dados Janela Ajuda

Naveg... Projetos *<despesas.db> Script-1

```

-- Valiando as categorias e subcategorias distintas
SELECT DISTINCT CATEGORIA, SUBCATEGORIA
FROM despesas
ORDER BY 1, 2;

-- Atualizando categorias e descrições específicas
UPDATE despesas
SET CATEGORIA = 'Beleza'
WHERE SUBCATEGORIA = 'Condicionador Colorido Cabelo';

● UPDATE despesas
SET CATEGORIA = 'Assinatura',
SUBCATEGORIA = 'iFood Club',
DESCRICAO = 'Assinatura iFood Club'
WHERE DESCRICAO = 'iFood Club';

● UPDATE despesas
SET CATEGORIA = 'Assinatura',
SUBCATEGORIA = 'Wellhub Academia',
DESCRICAO = 'Plano Academia Wellhub Jeff Cândido'
WHERE DESCRICAO = 'Jeff Cândido';

● UPDATE despesas
SET CATEGORIA = 'Assinatura',
SUBCATEGORIA = 'Kindle Unlimited'
WHERE DESCRICAO = 'Kindle Unlimited Amazon';

● Adicionando nova coluna de recorrência
ALTER TABLE despesas ADD COLUMN RECORRENCA TEXT;

● Preparando a coluna de recorrência
UPDATE despesas
SET RECORRENCA = 'Mensal'
WHERE CATEGORIA = 'Assinatura'
OR DESCRICAO = 'Mensalidade Faculdade Estácio';

● UPDATE despesas
SET RECORRENCA = 'Pontual'
WHERE RECORRENCA IS NULL;

● Inserindo novos dados manualmente
INSERT INTO despesas (DATA, DESCRICAO, CATEGORIA, SUBCATEGORIA, VALOR, RECORRENCA)
VALUES
('2025-01-13', 'Imposto Motina IPA', 'Transporte', 'IPA', '479,56', 'Anual'),
('2024-12-04', 'Crunchyroll Tikito', 'Assinatura', 'Crunchyroll', '50,00', 'Anual'),
('2024-11-25', 'Eppi Cinema Tikito', 'Assinatura', 'Eppi Cinema', '86,00', 'Anual'),
('2025-02-10', 'Spotify Rodolfo', 'Assinatura', 'Spotify', '56,00', 'Anual');

-- Removendo a informação de horário na coluna DATA
UPDATE despesas
SET DATA = DATE(DATA);

● Verificando a tabela final
SELECT * FROM despesas;

```

despesas 1

Ativar Salvar Cancelar Exportar dados 200 200+ BRT pt_BR Gravável Inserção Inteligente 55:1:1691

DBeaver 25.0.4 - <despesas.db> Script-1

Arquivo Editar Navegar Procurar Editor SQL Banco de dados Janela Ajuda

Naveg... Projetos *<despesas.db> Script-1

```

-- Removendo a informação de horário na coluna DATA
UPDATE despesas
SET DATA = DATE(DATA);

● Verificando a tabela final
SELECT * FROM despesas;

```

despesas 1

Größe AZ DATA AZ DESCRIÇÃO AZ CATEGORIA AZ SUBCATEGORIA I23 VALOR AZ RECORRENCA

1	2026-05-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19,9	Mensal
2	2026-05-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148,24	Mensal
3	2026-05-10	Assinatura iFood Club	Assinatura	iFood Club	5,95	Mensal
4	2026-05-10	Plano Academia Wellhub Jeff Cândido	Assinatura	Wellhub Academia	60	Mensal
5	2026-04-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19,9	Mensal
6	2026-04-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148,24	Mensal
7	2026-04-10	Assinatura iFood Club	Assinatura	iFood Club	5,95	Mensal
8	2026-04-10	Plano Academia Wellhub Jeff Cândido	Assinatura	Wellhub Academia	60	Mensal
9	2026-04-09	Roupas Micaelly Mih Store	Vestuário	Roupas	13,03	Pontual
10	2026-04-09	MSN Indústria e Comércio de Roupas	Vestuário	Roupas	52,58	Pontual
11	2026-03-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19,9	Mensal
12	2026-03-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148,24	Mensal
13	2026-03-10	Assinatura iFood Club	Assinatura	iFood Club	5,95	Mensal
14	2026-03-10	Plano Academia Wellhub Jeff Cândido	Assinatura	Wellhub Academia	60	Mensal
15	2026-03-09	Roupas Micaelly Mih Store	Vestuário	Roupas	13,03	Pontual
16	2026-03-09	MSN Indústria e Comércio de Roupas	Vestuário	Roupas	52,58	Pontual
17	2026-03-08	Roupas Micaelly Mih Store	Vestuário	Roupas	26,8	Pontual
18	2026-03-01	Amazon Livros	Lazer	Leitura	36,43	Pontual
19	2026-03-01	Shopee Lojinhaian	Lazer	Papelaria	26,56	Pontual
20	2026-02-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19,9	Mensal
21	2026-02-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148,24	Mensal
22	2026-02-10	Assinatura iFood Club	Assinatura	iFood Club	5,95	Mensal
23	2026-02-10	Plano Academia Wellhub Jeff Cândido	Assinatura	Wellhub Academia	60	Mensal
24	2026-02-09	Roupas Micaelly Mih Store	Vestuário	Roupas	13,03	Pontual
25	2026-02-09	MSN Indústria e Comércio de Roupas	Vestuário	Roupas	52,58	Pontual
26	2026-02-08	Roupas Micaelly Mih Store	Vestuário	Roupas	26,8	Pontual
27	2026-02-01	Amazon Livros	Lazer	Leitura	36,43	Pontual
28	2026-02-01	Amazon Livros	Lazer	Leitura	14,36	Pontual
29	2026-02-01	Drogaria Ultra Popular	Saúde	Medicamentos	4,5	Pontual
30	2026-02-01	Amazon Livros	Lazer	Leitura	12,38	Pontual
31	2026-02-01	Shopee Lojinhaian	Lazer	Papelaria	26,56	Pontual
32	2026-02-01	Usevitai	Vestuário	Sapatos	46,3	Pontual

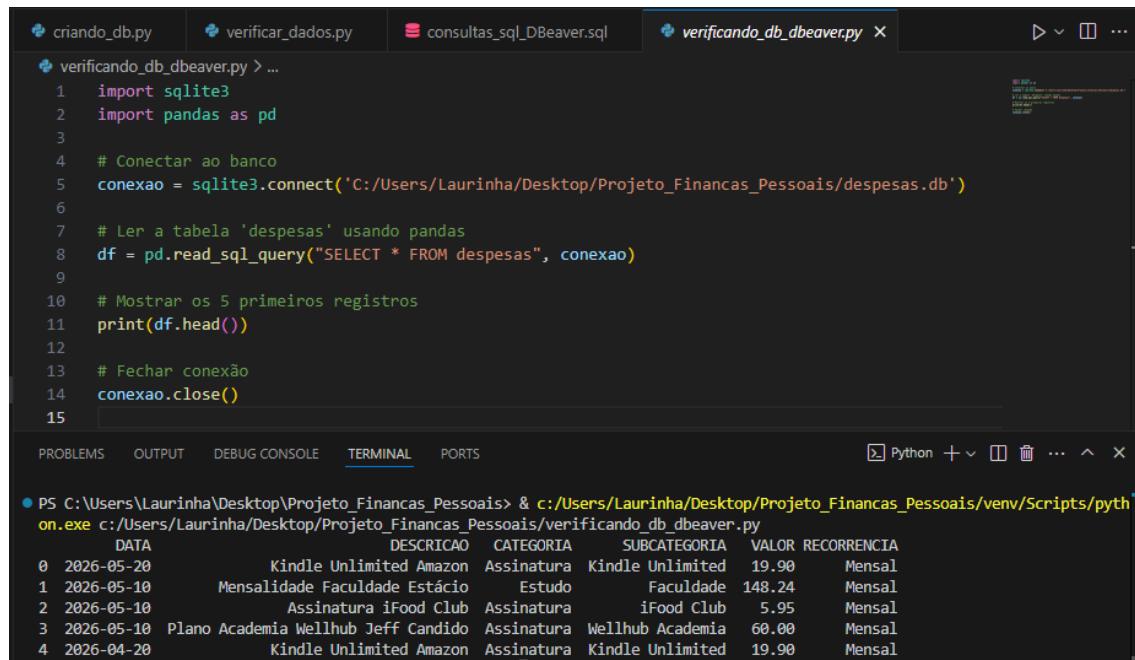
200 linha(s) recuperada(s) - 0,001s (0,001s recuperado), em 2025-05-05 às 14:16:17

Atualizar Salvar Cancelar Exportar dados 200 200+ BRT pt_BR Gravável Inserção Inteligente 55:1:1691

9. VALIDAÇÃO FINAL DO BANCO DE DADOS NO VS CODE

Após as transformações executadas no DBeaver (ajustes de categorias, adição de nova coluna e inserção de dados manuais), foi realizada uma verificação no VSCode para garantir que os dados estavam corretamente estruturados e salvos no banco.

Foi utilizado o script ‘verificando_db_dbeaver.py’, com a leitura da tabela despesas diretamente do banco despesas.db, exibindo os primeiros registros com o head() do pandas.



The screenshot shows the VS Code interface with several tabs open: 'criando_db.py', 'verificar_dados.py', 'consultas_sql_DBeaver.sql', and 'verificando_db_dbeaver.py'. The 'verificando_db_dbeaver.py' tab is active, displaying Python code. Below the tabs is a terminal window showing the command 'python c:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais/verificando_db_dbeaver.py' and its output, which is a table of transaction data from the 'despesas' table.

	DATA	DESCRICAÇÃO	CATEGORIA	SUBCATEGORIA	VALOR	RECORRENÇIA
0	2026-05-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19.90	Mensal
1	2026-05-10	Mensalidade Faculdade Estácio	Estudo	Faculdade	148.24	Mensal
2	2026-05-10	Assinatura iFood Club	Assinatura	iFood Club	5.95	Mensal
3	2026-05-10	Plano Academia Wellhub Jeff Candido	Assinatura	Wellhub Academia	60.00	Mensal
4	2026-04-20	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19.90	Mensal

10. CRIAÇÃO DO CONTAINER METABASE NO DOCKER USANDO O POWERSHELL

Descrição da criação do ambiente Metabase via Docker para disponibilizar visualizações interativas.

Etapas realizadas:

1. Criação inicial do container para testes:

```
docker run -d -p 3000:3000 --name metabase metabase/metabase
```

2. Parar e remover container anterior:

```
docker stop metabase
```

```
docker rm metabase
```

3. Criação do container definitivo com persistência de dados e acesso ao projeto:

```
docker run -d -p 3000:3000 --name metabase_seguro -v "C:/Users/Laurinha/Desktop/MetabaseData:/metabase-data" -v "C:/Users/Laurinha/Desktop/Projeto_Financas_Pessoais:/projeto" -e "MB_DB_FILE=/metabase-data/metabase.db" metabase/metabase
```

4. Listar containers e volumes para conferência:

```
docker ps -a
```

```
docker volume ls
```

A configuração com volumes garante que tanto o banco de dados do projeto (despesas.db) quanto o banco interno do Metabase (metabase.db, responsável por guardar usuários, perguntas, dashboards, etc.) sejam preservados mesmo que o container seja parado ou removido.

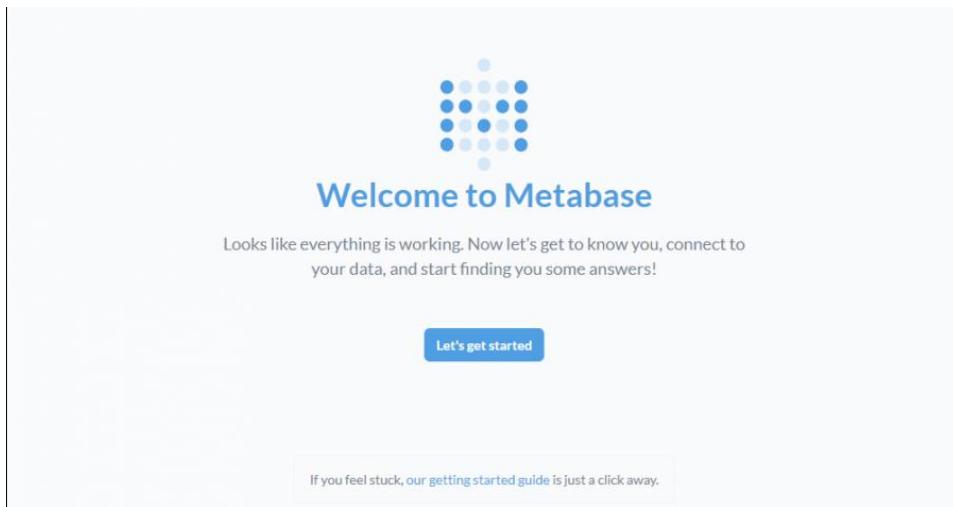
Isso foi possível ao mapear pastas locais no computador para o container usando o parâmetro -v. Dessa forma, toda a estrutura visual e lógica do Metabase permanece salva na pasta C:/Users/Laurinha/Desktop/MetabaseData, e o arquivo SQLite acessado permanece disponível na pasta Projeto_Financas_Pessoais.

11. CONECTANDO O BANCO DE DADOS SQLITE NO METABASE

Após a criação e execução do container do Metabase com os volumes devidamente mapeados, foi realizada a conexão entre o Metabase e o banco de dados despesas.db, que contém todas as informações organizadas para análise.

Passos realizados:

- Acessei a interface do Metabase pelo navegador, em <http://localhost:3000>.



- No primeiro acesso, fiz as configurações iniciais como idioma, usuário e senha, conforme solicitado pela plataforma e loguei com os dados criados.

The image consists of three vertically stacked screenshots of the Metabase application interface:

- Top Screenshot:** A configuration step titled "Qual é seu idioma preferido?" (What is your preferred language?). It shows a list of languages with "Portuguese" selected. Below the list is a "Próximo" (Next) button.
- Middle Screenshot:** The login screen titled "Entre no Metabase". It features a lighthouse illustration on the left and a bridge illustration on the right. The form includes fields for "Endereço de e-mail" (Email address) containing "voceestaincrivelhoje@email.com", "Senha" (Password) with placeholder "Shhh...", and a "Lembre-me" (Remember me) checkbox. Below the form is a link "Parece que esqueci minha senha" (Forgot my password?).
- Bottom Screenshot:** The main Metabase dashboard at the URL "localhost:3000". The left sidebar has a dark theme with navigation items like "Início", "Como usar o Metabase", "COLEÇÕES" (Collections) with "Nossas análises" (Our analyses) expanded, "Sua coleção pessoal" (Personal collection), "Examples", and "NAVEGAR" (Navigate) with "Modelos", "Bancos de Dados", "Métricas", and "Lixeira". The main area displays a greeting "Hey, Laura", a message "Continue de onde parou" (Continue where you left off), and four cards: "Mewlhão", "Despesas", "Despesas por Categoria", and "Metabase dicas". A "Pesquisar" (Search) bar and a "Novo" (New) button are at the top right.

- Na etapa de conexão, do lado esquerdo da tela cliquei em “Bancos de Dados” selecionei a opção Adicionar um banco de dados.

Início

Como usar o Metabase

COLEÇÕES

Nossas análises

Sua coleção pessoal

Examples

NAVEGAR

Modelos

Bancos de Dados

Métricas

Bancos de Dados

Aprenda sobre seus dados

Mewlhão

Sample Database

Adicionar um banco de dados

Mais de 20 conectores de dados. Comece a explorar em minutos.

- Informei o caminho do arquivo do banco de dados da seguinte forma: /projeto/despesas.db (Esse caminho corresponde ao diretório mapeado no container (via parâmetro -v) para a pasta do projeto no computador).

Adicionar um banco de dados

Tipo do banco de dados

SQLite

Nome amigável

Mewlhão

Filename

/projeto/despesas.db

Mostrar opções avançadas

Precisa de ajuda para conectar?

Cancelar

Salvar

- Finalizei a configuração e aguardei o carregamento da tabela. O Metabase reconheceu automaticamente a estrutura da base o que me permitiu criar dashboards, perguntas e visualizações interativas.

Início

Como usar o Metabase

COLEÇÕES

Nossas análises

Sua coleção pessoal

Examples

NAVEGAR

Modelos

Bancos de Dados

Métricas

Bancos de Dados

Aprenda sobre seus dados

BANCOS DE DADOS > MEWLHÃO

Despesas

Data	Descrição	Categoria	Subcategoria	Valor	Recorrência
maio 20, 2026, 12:00 AM	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19.9	Mensal
maio 10, 2026, 12:00 AM	Mensalidade Faculdade Estácio	Estudo	Faculdade	148.24	Mensal
maio 10, 2026, 12:00 AM	Assinatura iFood Club	Assinatura	iFood Club	5.95	Mensal
maio 10, 2026, 12:00 AM	Plano Academia Wellhub Jeff Candido	Assinatura	Wellhub Academia	60	Mensal
abril 20, 2026, 12:00 AM	Kindle Unlimited Amazon	Assinatura	Kindle Unlimited	19.9	Mensal
abril 10, 2026, 12:00 AM	Mensalidade Faculdade Estácio	Estudo	Faculdade	148.24	Mensal
abril 10, 2026, 12:00 AM	Assinatura iFood Club	Assinatura	iFood Club	5.95	Mensal
abril 10, 2026, 12:00 AM	Plano Academia Wellhub Jeff Candido	Assinatura	Wellhub Academia	60	Mensal

Observação: Caso o banco de dados seja alterado fora do Metabase (como por meio de scripts Python), as mudanças aparecerão automaticamente nas próximas atualizações de visualização, desde que o arquivo despesas.db seja mantido com o mesmo nome e no mesmo caminho mapeado.

12. CRIANDO DASHBOARD NO METABASE

Com os dados já estruturados no banco de dados SQLite e o Metabase configurado via Docker, foi possível construir um painel interativo para análise das despesas pessoais.

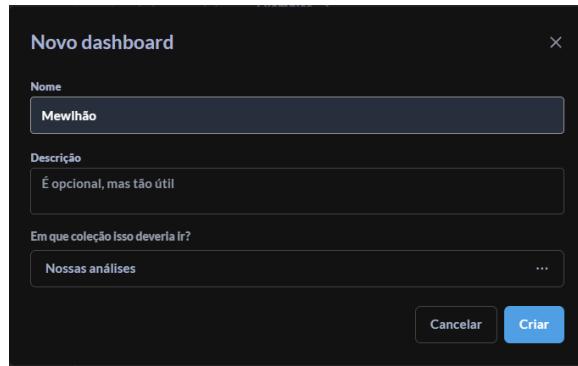
O dashboard reúne visualizações estratégicas criadas a partir de consultas SQL personalizadas e perguntas com filtros interativos.

Para iniciar a criação:

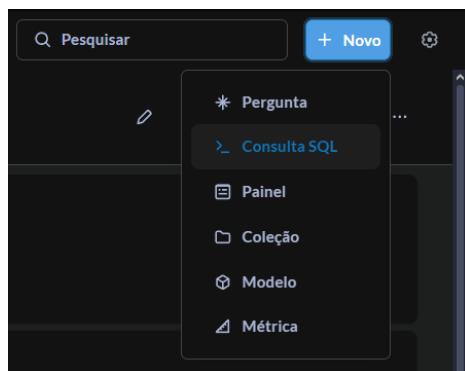
- No menu lateral do Metabase, cliquei em "Nossas análises", depois em "+ Novo" > "Painel".

The screenshot shows the Metabase interface with the title "Nossas análises". On the left, there's a sidebar with links like "Início", "Como usar o Metabase", "COLEÇÕES", and "NAVEGAR". The "COLEÇÕES" section has a sub-menu with "Nossas análises" (which is highlighted in blue) and other options like "Sua coleção pessoal" and "Examples". On the right, there's a table with columns "Tipo", "Nome", and "Último por". A context menu is open on the right, with "Painel" selected. Other options in the menu include "Pergunta", "Consulta SQL", "Coleção", "Modelo", and "Métrica".

- Preenchi o nome do dashboard e selecionei a coleção desejada.



- Após clicar em “Criar”, com o painel aberto, selecionei “+ Novo” > “Consulta SQL” para adicionar visualizações baseadas em código SQL.



Ao todo, foram criadas 7 visualizações SQL e 1 pergunta com filtros. Abaixo estão os gráficos desenvolvidos:

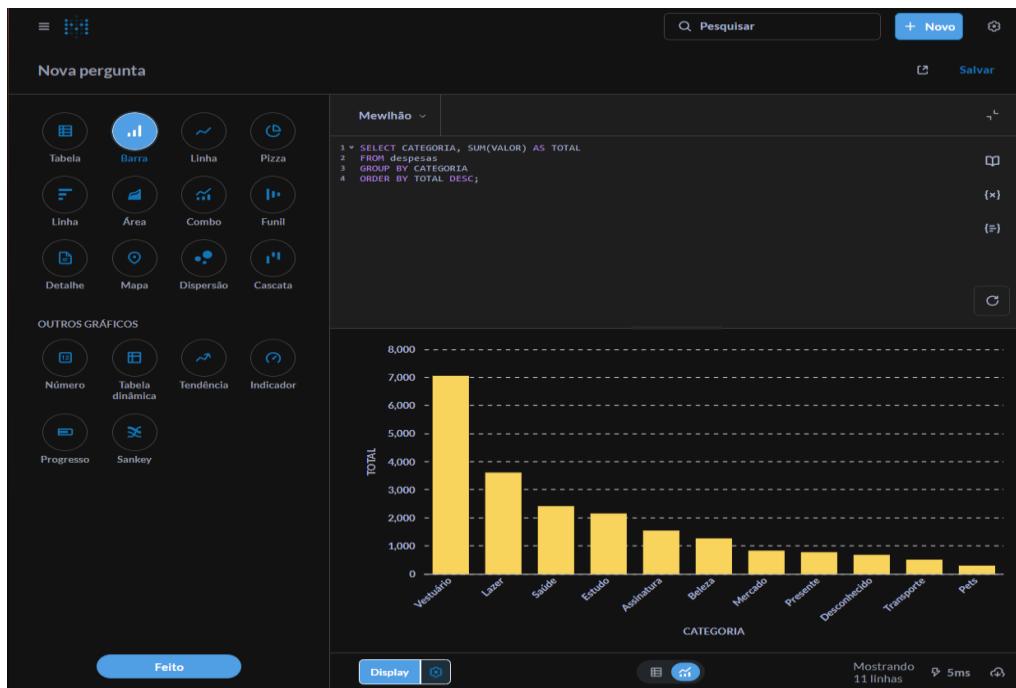
12.1. Total de Despesas (Número)

Consulta que soma o valor total de todas as despesas. A visualização foi configurada como "Número".

O mesmo processo foi repetido com os demais gráficos:

12.2. Despesas por Categoria (Gráfico de Barras)

Consulta que agrupa as despesas por categoria, somando os valores e ordenando do maior para o menor. Visualização em gráfico de barras verticais.



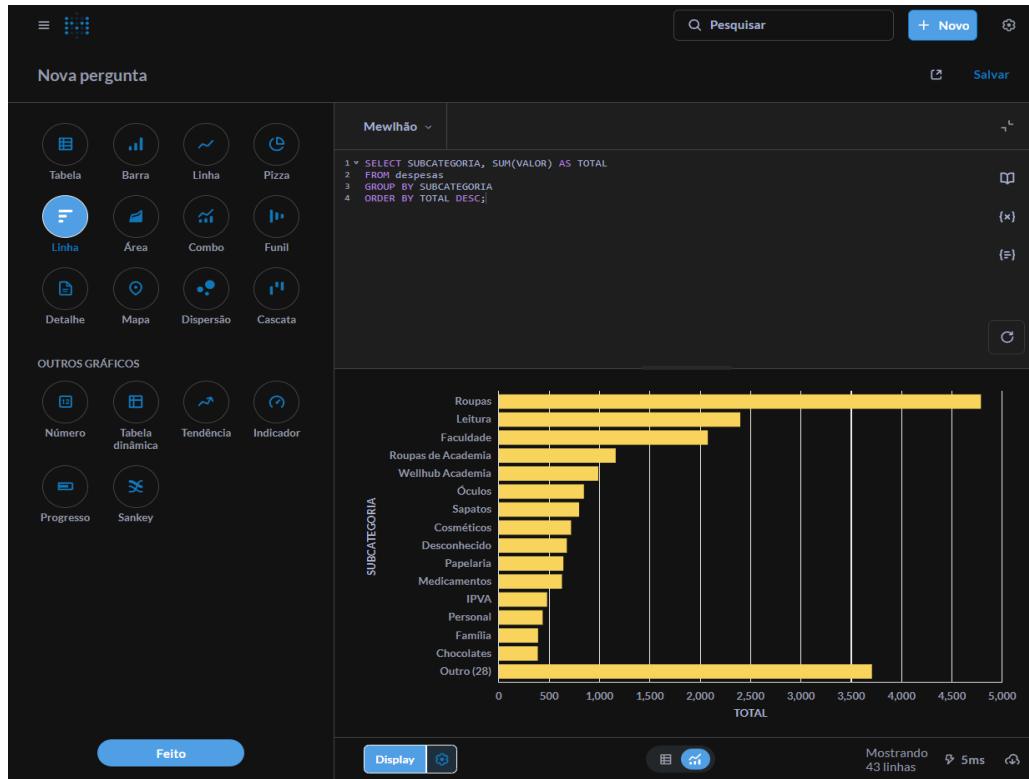
12.3. Despesas por Categoria - Porcentagem (Gráfico de Pizza)

Variação da anterior, mas com a visualização em gráfico de pizza para representar a proporção de cada categoria.



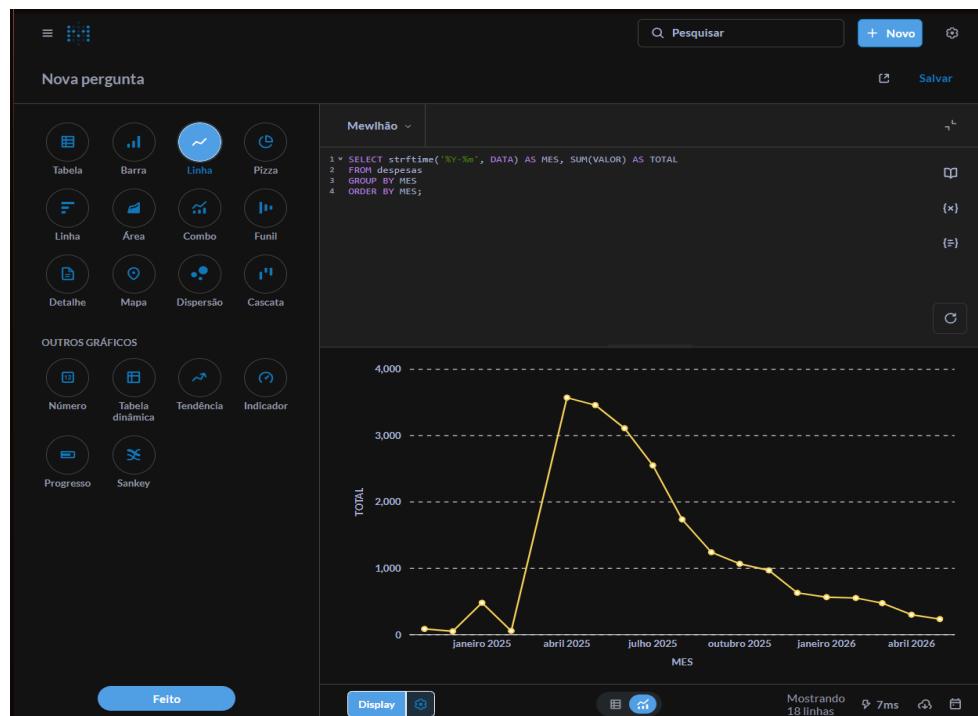
12.4. Despesas por Subcategoria (Gráfico de Barras)

Consulta semelhante à de categorias, mas detalhando por subcategoria. Ideal para entender com mais granularidade onde os gastos estão concentrados.



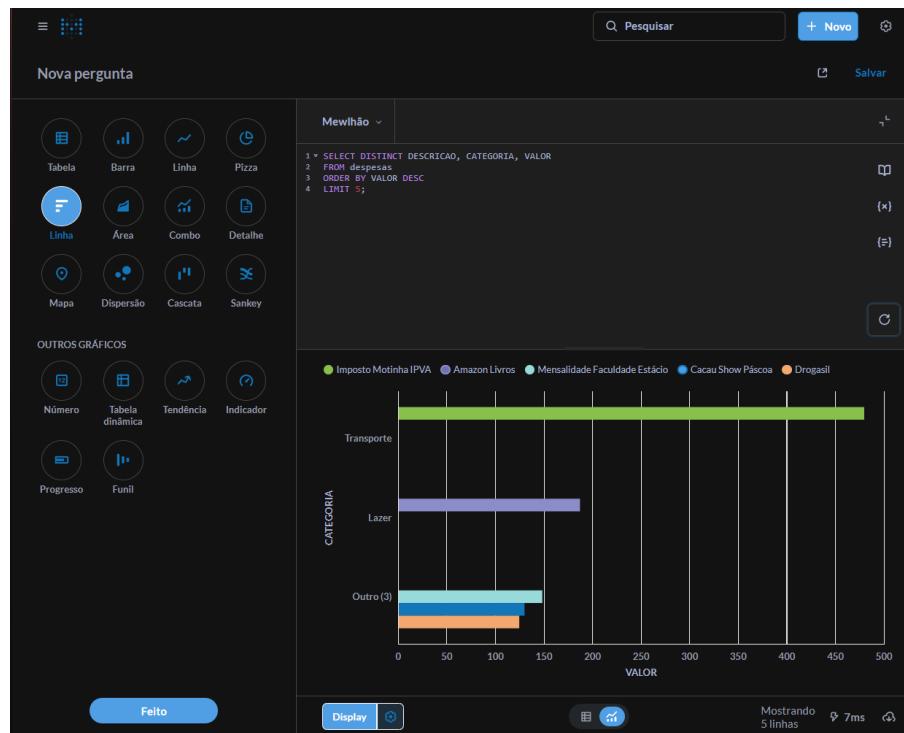
12.5. Despesas por Mês (Gráfico de Linha)

Soma das despesas agrupadas por mês (formato YYYY-MM). Representada em gráfico de linha para facilitar a visualização de tendências ao longo do tempo.



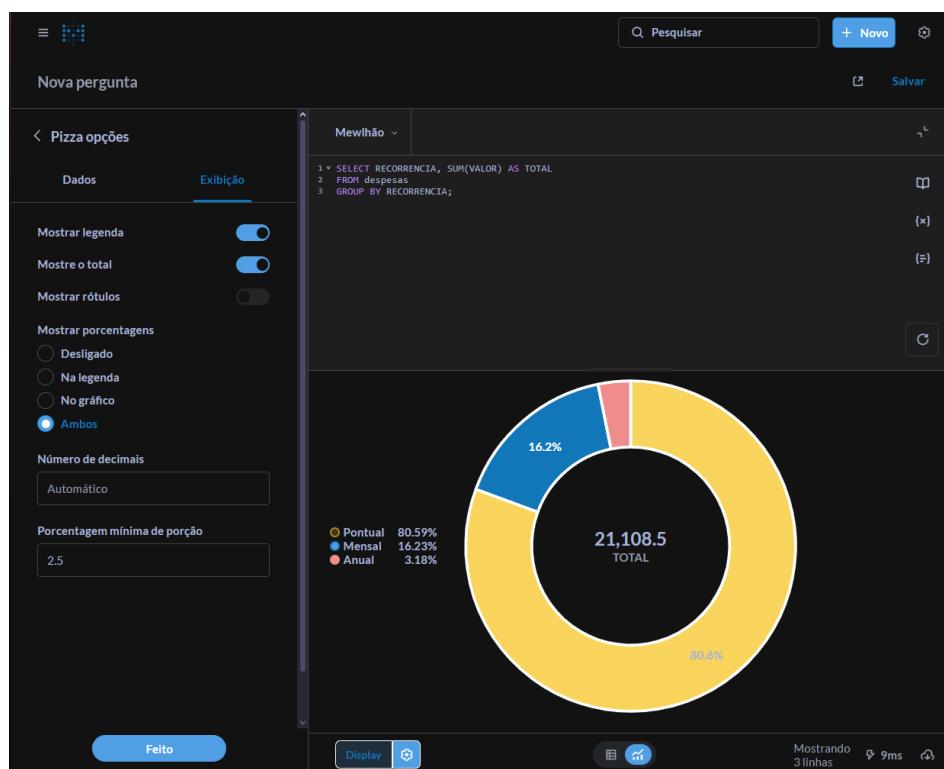
12.6. Top 5 Maiores Despesas (Tabela ou Gráfico Horizontal)

Lista das 5 maiores despesas do período, com valores e descrição. Pode ser visualizado como tabela ou gráfico de colunas horizontais.



12.7. Despesas Recorrentes vs Únicas (Gráfico de Pizza)

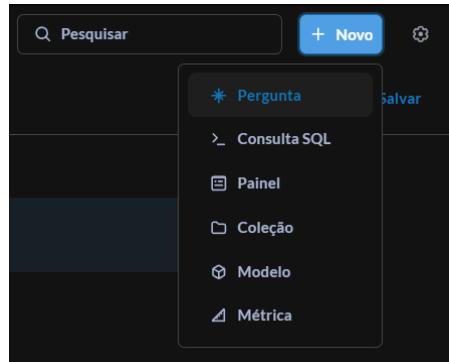
Agrupamento das despesas com base no campo “Recorrência”, dividido entre “Mensal”, “Pontual” e “Anual”. Visualização em gráfico de pizza para melhor proporção.



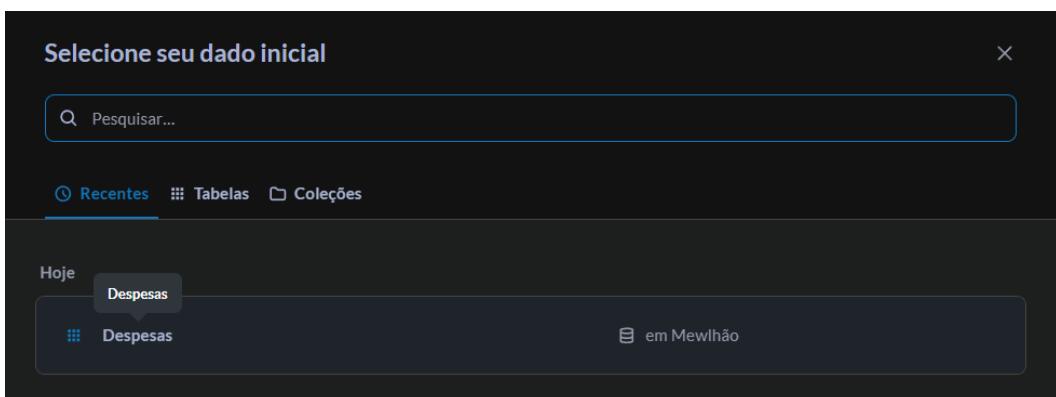
12.8. Tabela com Filtros (Pergunta no Dashboard)

Além das visualizações SQL, foi adicionada uma **pergunta interativa** ao painel:

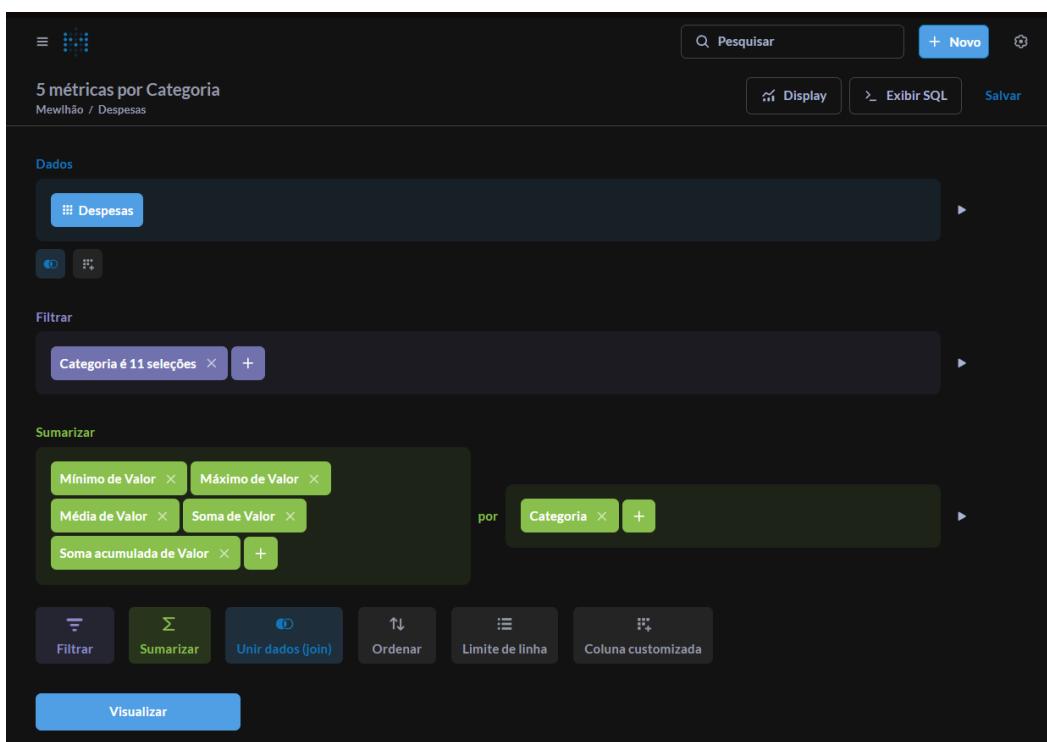
- Cliquei em “+ Novo” > “Pergunta”;



- Selecionei a tabela despesas do banco de dados mewlhão;



- Adicionei os filtros desejados de valores;



- Escolhi a visualização em tabela e salvei no painel.

Categoria	Mínimo de Valor	Máximo de Valor	Média de Valor	Soma de Valor	Soma acumulada de Valor
Assinatura	5.95	86	28.03	1.541.8	1.541.8
Beleza	8.03	72.55	25.26	1.263.07	2.804.87
Desconhecido	10.93	57.73	32.16	675.44	3.480.31
Estudo	15.27	148.24	113.25	2.151.71	5.632.02
Lazer	12.38	187.02	34.71	3.609.97	9.241.99
Mercado	5.02	68.82	22.3	825.03	10.067.02
Pets	29.18	29.18	29.18	291.8	10.358.82
Presente	16.15	129.92	45.34	770.72	11.129.54
Saúde	4.5	124.49	34.98	2.413.63	13.543.17
Transporte	25	479.56	252.28	504.56	14.047.73
Vestuário	13.03	123.18	47.39	7.060.77	21.108.5

Essa tabela permite que o usuário final filtre os dados em tempo real, tornando a análise mais dinâmica e exploratória.

13. CONSIDERAÇÕES FINAIS

Este projeto teve como objetivo desenvolver um painel interativo para o controle e análise de finanças pessoais, utilizando ferramentas amplamente acessíveis e gratuitas. O fluxo completo — da coleta de dados em CSV, organização via SQLite, até a visualização interativa com Metabase — demonstrou como é possível criar soluções eficientes e visuais mesmo com recursos simples.

Durante o processo, foram aplicados conceitos de modelagem de dados, consulta SQL, manipulação de arquivos com PowerShell e utilização de contêineres Docker, promovendo uma experiência prática em Engenharia de Dados. O uso do Metabase possibilitou uma interface amigável para criação de dashboards que facilitam a tomada de decisões financeiras de forma visual e acessível.

Além disso, a persistência dos dados em volumes externos garantiu a segurança e continuidade do projeto mesmo em caso de reinicialização ou exclusão do container Docker.

Com isso, conclui-se que a união de boas práticas em organização de dados, automação e visualização pode transformar informações simples em insights valiosos para o dia a dia.

14. REFERÊNCIAS

- [Metabase – Site Oficial](#)
- Docker – Documentação Oficial
- SQLite – Site Oficial
- Documentação SQL (W3Schools)
- [PowerShell – Microsoft Learn](#)
- Cursos e vídeos utilizados para consulta durante o projeto:
 - Curso "Linguagem SQL para Análise de Dados" – Udemy
 - Vídeos e fóruns da comunidade (Stack Overflow, YouTube)