

✓ Projeto de Análise de Dados

Desenvolvido por Ian Rodrigues dos Reis Paixão

Contato: iang.master100@gmail.com

O objetivo deste projeto é validar a integridade do Balancete de um grupo econômico, considerando as movimentações contábeis organizadas por ativo e passivo.

As etapas do problema incluem:

- 1 **1** Limpeza e preparação de dados: Importar dados de diversas planilhas CSV, tratar inconsistências, atualizar valores nulos e remover duplicados.
- 2 **2** Análise temporal: Verificar a evolução do Balancete, observando o comportamento do ativo e do passivo.
- 3 **3** Análise por categoria: Consolidar o Balancete em categorias de ativo e passivo, garantindo a integridade do grupo econômico.
- 4 **4** Exportação de resultados: Gerar um relatório em formato JSON com a análise do Balancete.
- 5 **5** Visualização: Exibir os resultados de maneira interativa utilizando a biblioteca 'json2html'.

✨ Resultado: Um fluxo automatizado e confiável, ideal para auditorias financeiras, oferecendo uma solução eficiente e elegante

```

1 '''
2 -- Exercício 1: Modelagem e Tratamento de Dados
3
4 -- A) Criação do Diagrama de Entidade e Relacionamento (DER)
5
6 CREATE TABLE Empresa (
7    Codigo INT PRIMARY KEY,
8     Empresa VARCHAR(255) NOT NULL
9 );
10
11 CREATE TABLE MovimentoContabil (
12     Data DATE NOT NULL,
13     `Conta Contábil` VARCHAR(50) NOT NULL,
14     Descricao VARCHAR(255) NOT NULL,
15     Valor DECIMAL(15,2) NOT NULL,
16     Relatorio INT NOT NULL,
17     `Centro de Custo` INT NOT NULL,
18     Empresa INT NOT NULL,
19     Seq INT PRIMARY KEY
20 );
21
22 CREATE TABLE Relatorio (
23     Código INT PRIMARY KEY,
24     Relatorio VARCHAR(255) NOT NULL,
25     `Indicador Ativo / Inativo` VARCHAR(10) NOT NULL
26 );
27
28 CREATE TABLE CentroCusto (
29     Código INT PRIMARY KEY,
30     Descrição VARCHAR(255) NOT NULL,
31     `Indicador Ativo / Inativo` VARCHAR(10) NOT NULL
32 );
33
34 -- B) Tratamento dos dados
35
36 -- Atualizar valores nulos para zero nas tabelas necessárias
37 UPDATE MovimentoContabil
38 SET Valor = 0
39 WHERE Valor IS NULL;
40
41 -- Remover registros duplicados na MovimentoContabil
42 DELETE mc1
43 FROM MovimentoContabil mc1
44 INNER JOIN MovimentoContabil mc2
45     ON mc1.Seq > mc2.Seq
46     AND mc1.`Conta Contábil` = mc2.`Conta Contábil`
47     AND mc1.Empresa = mc2.Empresa;
48
49 -- C) Scripts para popular as tabelas conforme o arquivo Excel

```

```

50
51 -- Cada Planilha do excel foi Salva como um arquivo csv separadamente
52
53 -- Carregar dados na tabela Empresa
54 LOAD DATA INFILE 'C:/Users/iangm/Downloads/Empresa.csv'
55 INTO TABLE Empresa
56 FIELDS TERMINATED BY ','
57 LINES TERMINATED BY '\n'
58 IGNORE 1 ROWS;
59
60 -- Carregar dados na tabela MovimentoContabil
61 LOAD DATA INFILE 'C:/Users/iangm/Downloads/Movimento Contábil.csv'
62 INTO TABLE MovimentoContabil
63 FIELDS TERMINATED BY ','
64 LINES TERMINATED BY '\n'
65 IGNORE 1 ROWS;
66
67 -- Carregar dados na tabela Relatorio
68 LOAD DATA INFILE 'C:/Users/iangm/Downloads/Relatório.csv'
69 INTO TABLE Relatorio
70 FIELDS TERMINATED BY ','
71 LINES TERMINATED BY '\n'
72 IGNORE 1 ROWS;
73
74 -- Carregar dados na tabela CentroCusto
75 LOAD DATA INFILE 'C:/Users/iangm/Downloads/Centro de Custo.csv'
76 INTO TABLE CentroCusto
77 FIELDS TERMINATED BY ','
78 LINES TERMINATED BY '\n'
79 IGNORE 1 ROWS;
80
81 -- Exercício 2: Análise e Exportação para JSON
82
83 -- A) Validação da integridade do Balancete considerando Ativo e Passivo do grupo econômico
84
85 SELECT
86     SUM(CASE WHEN LEFT(`Conta Contábil`, 1) = '1' THEN Valor ELSE 0 END) AS total_ativo,
87     SUM(CASE WHEN LEFT(`Conta Contábil`, 1) != '1' THEN Valor ELSE 0 END) AS total_passivo,
88     (SUM(CASE WHEN LEFT(`Conta Contábil`, 1) = '1' THEN Valor ELSE 0 END) +
89     SUM(CASE WHEN LEFT(`Conta Contábil`, 1) != '1' THEN Valor ELSE 0 END)) AS integridade
90 FROM MovimentoContabil;
91
92 -- Regra: Integridade é quando TOTAL ATIVO + TOTAL PASSIVO = 0
93
94 -- B) Consulta para retornar JSON com a estrutura solicitada para o grupo econômico
95
96 SELECT
97     JSON_OBJECT(
98         'total_ativo', SUM(CASE WHEN LEFT(`Conta Contábil`, 1) = '1' THEN Valor ELSE 0 END),
99         'total_passivo', SUM(CASE WHEN LEFT(`Conta Contábil`, 1) != '1' THEN Valor ELSE 0 END),
100         'integridade', (SUM(CASE WHEN LEFT(`Conta Contábil`, 1) = '1' THEN Valor ELSE 0 END) +
101             SUM(CASE WHEN LEFT(`Conta Contábil`, 1) != '1' THEN Valor ELSE 0 END))
102     ) AS resultado_json
103 FROM MovimentoContabil;
104
105 '''

```

```

-- Exercício 1: Modelagem e Tratamento de Dados
-- A) Criação do Diagrama de Entidade e Relacionamento (DER)
CREATE TABLE Empresa (
    Codigo INT PRIMARY KEY,
    Empresa VARCHAR(255) NOT NULL
);
CREATE TABLE MovimentoContabil (
    Data DATE NOT NULL,
    `Conta Contábil` VARCHAR(50) NOT NULL,
    Descricao VARCHAR(255) NOT NULL,
    Valor DECIMAL(15,2) NOT NULL,
    Relatorio INT NOT NULL,
    `Centro de Custo` INT NOT NULL,
    Empresa VARCHAR(255) NOT NULL,
    Seq INT PRIMARY KEY
);
CREATE TABLE Relatorio (
    Código INT PRIMARY KEY,
    Relatorio VARCHAR(255) NOT NULL,
    `Indicador Ativo / Inativo` VARCHAR(10) NOT NULL
);
CREATE TABLE CentroCusto (
    Código INT PRIMARY KEY,
    Descrição VARCHAR(255) NOT NULL,
    `Indicador Ativo / Inativo` VARCHAR(10) NOT NULL
);
-- B) Tratamento dos dados
UPDATE MovimentoContabil

```

```

1 import pandas as pd
2 import sqlite3
3 import json
4 import IPython.display as display
5 from json2html import *
6
7 # Caminhos para os arquivos CSV no Google Colab
8 empresa_csv = '/content/Empresa.csv'
9 movimento_contabil_csv = '/content/Movimento Contábil.csv'
10 relatorio_csv = '/content/Relatório.csv'
11 centro_custo_csv = '/content/Centro de Custo.csv'
12
13 # Criando o banco de dados SQLite
14 conn = sqlite3.connect('/content/balancete.db')
15 cursor = conn.cursor()
16
17 # Criando as tabelas SQLite

```

```

18 cursor.execute('''
19     CREATE TABLE IF NOT EXISTS Empresa (
20         Codigo INT PRIMARY KEY,
21         Empresa TEXT NOT NULL
22     );
23 ''')
24
25 cursor.execute('''
26     CREATE TABLE IF NOT EXISTS MovimentoContabil (
27         Data TEXT NOT NULL,
28         "Conta Contábil" TEXT NOT NULL,
29         Descricao TEXT NOT NULL,
30         Valor REAL NOT NULL,
31         Relatorio INT NOT NULL,
32         "Centro de Custo" INT NOT NULL,
33         Empresa INT NOT NULL,
34         Seq INT PRIMARY KEY
35     );
36 ''')
37
38 cursor.execute('''
39     CREATE TABLE IF NOT EXISTS Relatorio (
40         Código INT PRIMARY KEY,
41         Relatorio TEXT NOT NULL,
42         "Indicador Ativo / Inativo" TEXT NOT NULL
43     );
44 ''')
45
46 cursor.execute('''
47     CREATE TABLE IF NOT EXISTS CentroCusto (
48         Código INT PRIMARY KEY,
49         Descrição TEXT NOT NULL,
50         "Indicador Ativo / Inativo" TEXT NOT NULL
51     );
52 ''')
53
54 print("Tabelas criadas com sucesso!")
55
56 # Carregar os arquivos CSV para DataFrames
57 df_empresa = pd.read_csv(empresa_csv)
58 df_movimento_contabil = pd.read_csv(movimento_contabil_csv)
59 df_relatorio = pd.read_csv(relatorio_csv)
60 df_centro_custo = pd.read_csv(centro_custo_csv)
61
62 # Enviando os DataFrames para o banco de dados SQLite
63 df_empresa.to_sql('Empresa', conn, if_exists='replace', index=False)
64 df_movimento_contabil.to_sql('MovimentoContabil', conn, if_exists='replace', index=False)
65 df_relatorio.to_sql('Relatorio', conn, if_exists='replace', index=False)
66 df_centro_custo.to_sql('CentroCusto', conn, if_exists='replace', index=False)
67
68 print("Dados inseridos com sucesso!")
69
70 # Tratamento dos dados via SQL
71 cursor.execute('''
72     UPDATE MovimentoContabil
73     SET Valor = 0
74     WHERE Valor IS NULL;
75 ''')
76
77 cursor.execute('''
78     DELETE FROM MovimentoContabil
79     WHERE rowid NOT IN (
80         SELECT MIN(rowid)
81         FROM MovimentoContabil
82         GROUP BY "Conta Contábil", Empresa, Seq
83     );
84 ''')
85
86 conn.commit()
87 print("Tratamento dos dados realizado com sucesso!")
88
89 # Consulta SQL para gerar o JSON estruturado
90 query_json = """
91 SELECT
92     e.Empresa AS empresa,
93     SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) = '1' THEN mc.Valor ELSE 0 END) AS total_ativo,
94     SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) != '1' THEN mc.Valor ELSE 0 END) AS total_passivo,
95     (SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) = '1' THEN mc.Valor ELSE 0 END) +
96      SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) != '1' THEN mc.Valor ELSE 0 END)) AS check_integridade
97 FROM MovimentoContabil mc
98 JOIN Empresa e ON mc.Empresa = e.Codigo
99 GROUP BY e.Empresa;
100 """

```

```

101
102 df_json = pd.read_sql_query(query_json, conn)
103
104 # Converter para JSON
105 result_json = df_json.to_json(orient='records', indent=4)
106
107 # Salvar JSON
108 json_path = '/content/resultado_balancete.json'
109 with open(json_path, 'w') as f:
110     f.write(result_json)
111
112 print(f"Arquivo JSON gerado com sucesso! Caminho: {json_path}")
113
114 # Abrir o JSON em uma visualização HTML interativa
115 with open(json_path, 'r') as f:
116     json_content = json.load(f)
117     html_content = json2html.convert(json = json_content)
118     display.display(display.HTML(html_content))
119
120 # Fechar a conexão com o banco de dados
121 conn.close()
122
123

```

 Tabelas criadas com sucesso!
 Dados inseridos com sucesso!
 Tratamento dos dados realizado com sucesso!
 Arquivo JSON gerado com sucesso! Caminho: /content/resultado_balancete.json

empresa	total_ativo	total_passivo	check_integridade
EMPRESA 1	2025109140.4	12237427.98	2037346568.38
EMPRESA 2	184290832.0399999	56767969.66000001	241058801.69999993
EMPRESA 3	-7051.2	-9924192.03	-9931243.229999999
EMPRESA 4	1497706894.3999999	1160661.7199999997	1498867556.12

Double-click (or enter) to edit

```

1  # Autor: Ian Rodrigues
2
3  import pandas as pd
4  import sqlite3
5  import json
6  import IPython.display as display
7  from json2html import *
8
9  # Caminhos para os arquivos CSV no Google Colab
10 empresa_csv = '/content/Empresa.csv'
11 movimento_contabil_csv = '/content/Movimento Contábil.csv'
12 relatorio_csv = '/content/Relatório.csv'
13 centro_custo_csv = '/content/Centro de Custo.csv'
14
15 # Criando o banco de dados SQLite
16 conn = sqlite3.connect('/content/balancete.db')
17 cursor = conn.cursor()
18
19 # Criando as tabelas SQLite
20 cursor.execute('''
21     CREATE TABLE IF NOT EXISTS Empresa (
22         Código INT PRIMARY KEY,
23         Empresa TEXT NOT NULL
24     );
25 ''')
26
27 cursor.execute('''
28     CREATE TABLE IF NOT EXISTS MovimentoContabil (
29         Data TEXT NOT NULL,
30         "Conta Contábil" TEXT NOT NULL,
31         Descricao TEXT NOT NULL,
32         Valor REAL NOT NULL,
33         Relatorio INT NOT NULL,
34         "Centro de Custo" INT NOT NULL,
35         Empresa INT NOT NULL,
36         Seq INT PRIMARY KEY
37     );
38 ''')
39
40 cursor.execute('''
41     CREATE TABLE IF NOT EXISTS Relatorio (
42         Código INT PRIMARY KEY,
43         Relatorio TEXT NOT NULL,
44         "Indicador Ativo / Inativo" TEXT NOT NULL
45     );
46 ''')

```

```

45     );
46     '')
47
48     cursor.execute('''
49         CREATE TABLE IF NOT EXISTS CentroCusto (
50             Código INT PRIMARY KEY,
51             Descrição TEXT NOT NULL,
52             "Indicador Ativo / Inativo" TEXT NOT NULL
53         );
54     '')
55
56     print("Tabelas criadas com sucesso!")
57
58     # Carregar os arquivos CSV para DataFrames
59     df_empresa = pd.read_csv(empresa_csv)
60     df_movimento_contabil = pd.read_csv(movimento_contabil_csv)
61     df_relatorio = pd.read_csv(relatorio_csv)
62     df_centro_custo = pd.read_csv(centro_custo_csv)
63
64     # Enviando os DataFrames para o banco de dados SQLite
65     df_empresa.to_sql('Empresa', conn, if_exists='replace', index=False)
66     df_movimento_contabil.to_sql('MovimentoContabil', conn, if_exists='replace', index=False)
67     df_relatorio.to_sql('Relatorio', conn, if_exists='replace', index=False)
68     df_centro_custo.to_sql('CentroCusto', conn, if_exists='replace', index=False)
69
70     print("Dados inseridos com sucesso!")
71
72     # Tratamento dos dados via SQL
73     cursor.execute('''
74         UPDATE MovimentoContabil
75         SET Valor = 0
76         WHERE Valor IS NULL;
77     ''')
78
79     cursor.execute('''
80         DELETE FROM MovimentoContabil
81         WHERE rowid NOT IN (
82             SELECT MIN(rowid)
83             FROM MovimentoContabil
84             GROUP BY "Conta Contábil", Empresa, Seq
85         );
86     ''')
87
88     conn.commit()
89     print("Tratamento dos dados realizado com sucesso!")
90
91     # Consulta SQL para gerar o JSON estruturado
92     query_json = """
93     SELECT
94         e.Empresa AS empresa,
95         SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) = '1' THEN mc.Valor ELSE 0 END) AS total_ativo,
96         SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) != '1' THEN mc.Valor ELSE 0 END) AS total_passivo,
97         (SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) = '1' THEN mc.Valor ELSE 0 END) +
98          SUM(CASE WHEN substr(mc."Conta Contábil", 1, 1) != '1' THEN mc.Valor ELSE 0 END)) AS check_integridade
99     FROM MovimentoContabil mc
100    JOIN Empresa e ON mc.Empresa = e.Codigo
101    GROUP BY e.Empresa;
102    """
103
104     df_json = pd.read_sql_query(query_json, conn)
105
106     # Converter para JSON
107     result_json = df_json.to_json(orient='records', indent=4)
108
109     # Salvar JSON
110     json_path = '/content/resultado_balancete.json'
111     with open(json_path, 'w') as f:
112         f.write(result_json)
113
114     print(f"Arquivo JSON gerado com sucesso! Caminho: {json_path}")
115
116     # Abrir o JSON em uma visualização HTML interativa
117     with open(json_path, 'r') as f:
118         json_content = json.load(f)
119         html_content = json2html.convert(json = json_content)
120         display.display(display.HTML(html_content))
121
122     # Fechar a conexão com o banco de dados
123     conn.close()
124

```

➡ Tabelas criadas com sucesso!
Dados inseridos com sucesso!
Tratamento dos dados realizado com sucesso!
Arquivo JSON gerado com sucesso! Caminho: /content/resultado_balancete.json

empresa	total_ativo	total_passivo	check_integridade
EMPRESA 1	2025109140.4	12237427.98	2037346568.38
EMPRESA 2	184290832.03999999	56767969.66000001	241058801.69999993
EMPRESA 3	-7051.2	-9924192.03	-9931243.229999999
EMPRESA 4	1497706894.3999999	1160661.7199999997	1498867556.12
EMPRESA 5	1876694707.36	-323909775.46	1552784931.8999999