

Avaliação 2 de Introdução a Computação

Nathan Loose Kuipper
Rafael Gontijo Ferreira

5 de junho de 2025

Resumo

Resumo do trabalho

1 Introdução

Introdução do trabalho.

2 Análise do Código

Introdução da Análise do Código

Módulo Auxiliar A2

Módulo que contém funções para auxiliar no manejo da base de dados Path:

```
1 AUTORES = ['Nathan_Loose_Kuipper', 'Rafael_Gontijo_Ferreira']
2
3
4 import pandas as pd
5 import sqlite3
6 from pathlib import Path
7
8 PATH = Path(__file__).parent # bilheteria.db na mesma pasta que esse arquivo
9
10 def queryconn(database, query):
11     with sqlite3.connect(database) as conn:
12         cursor = conn.cursor()
13         cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
14         tables = cursor.fetchall()
15
16         df = pd.read_sql_query(query, conn)
17
18     return df
19
20 def carrega_tabela(database, tabela):
21
22     with sqlite3.connect(database) as conn:
23         query = f"SELECT * FROM {tabela}"
24         df = pd.read_sql_query(query, conn)
25     return df
26
27 def lista_tabelas(db_filename):
28
29     # Connect to the SQLite database
30     # conn = sqlite3.connect(db_filename)
31     with sqlite3.connect(db_filename) as conn:
```

```

32     cursor = conn.cursor()
33     cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
34     tables = cursor.fetchall()
35
36     table_row_counts = []
37     for table in tables:
38         table_name = table[0]
39         query = f"SELECT COUNT(*) FROM {table_name};"
40         cursor.execute(query)
41         row_count = cursor.fetchone()[0]
42         table_row_counts.append({"Table": table_name, "Row_Count": row_count})
43
44     return pd.DataFrame(table_row_counts)
45
46 if __name__ == '__main__':
47     print("Importe esse modulo para auxiliar com o manejo da base de dados!")

```

Questão 1

```

1 def questao1():
2
3     dsessao = a2.carrega_tabela(PATH / 'bilheteria.db', 'sessao')
4     dfsessao = dsessao.groupby(by=['filme_id'])['publico'].sum().reset_index()
5
6     dfilme = a2.carrega_tabela(PATH / 'bilheteria.db', 'filme')
7
8     map_titulo = lambda x: dfilme.loc[dfilme['id'] == x, 'titulo_original'].item()
9     dfsessao['filme_id'] = dfsessao['filme_id'].map(map_titulo).astype(str)
10
11     return dfsessao

```

Questão 2

```

1 def questao2():
2     dfilme = a2.carrega_tabela(PATH / 'bilheteria.db', 'filme')
3     dsessao = a2.carrega_tabela(PATH / 'bilheteria.db', 'sessao')
4
5     dfsessao = dsessao.groupby(by=['filme_id'])['publico'].sum().reset_index()
6
7     merged_df = dfilme.merge(dfsessao, left_on='id', right_on='filme_id', how='left')
8
9     merged_df['publico'] = merged_df['publico'].fillna(0)
10
11     paises = merged_df['pais_origem'].unique()
12     dic = {}
13
14     for pais in paises:
15         most_viewed_film = merged_df[merged_df['pais_origem'] == pais].sort_values(by='publico',
16                                         ascending=False).iloc[0]
17         dic[pais] = {
18             'nome': dfilme.loc[dfilme['id'] == most_viewed_film['filme_id'], 'titulo_original']
19             .item(),
20             'publico': int(most_viewed_film['publico'])
21         }
22
23     return dic

```

Questão 3

Questão 4

```
1 def questao4():
2
3     dsessao = a2.carrega_tabela(PATH / 'bilheteria.db', 'sessao')
4     dsala = a2.carrega_tabela(PATH / 'bilheteria.db', 'sala')[['id', 'from_complexo']]
5     dcomplexo = a2.carrega_tabela(PATH / 'bilheteria.db', 'complexo')[['id', 'municipio']]
6
7     dfilme = a2.carrega_tabela(PATH / 'bilheteria.db', 'filme')[['id', 'titulo_original']]
8
9     df = dsessao.merge(dsala, left_on='sala_id', right_on='id', how='left')
10    df = df.rename(columns={'id_x': 'sessao_id', 'id_y': 'sala_id'})
11
12    df = df.merge(dcomplexo, left_on='from_complexo', right_on='id', how='left')
13    df = df.rename(columns={'municipio': 'CIDADE'})
14
15    df = df.merge(dfilme, left_on='filme_id', right_on='id', how='left')
16    df = df.rename(columns={'titulo_original': 'FILME'})
17
18    bilheteria = df.groupby(['CIDADE', 'FILME'], as_index=False)['publico'].sum()
19    bilheteria = bilheteria.rename(columns={'publico': 'BILHETERIA'})
20
21    resultado = bilheteria.sort_values('BILHETERIA', ascending=False).groupby('CIDADE').head
22    (1)
23
24    return resultado[['CIDADE', 'FILME', 'BILHETERIA']]
```

Questão 5

Referências

[1] Autor, A. (Ano). *Título do Livro*. Editora.