

```

import pandas as pd
from collections import deque

def limpar_valor(valor):
    #Converte um valor monetario em texto para um numero
    if isinstance(valor, str):
        try:
            return float(valor.strip().replace('.', '').replace(',', '.'))
        except ValueError:
            return 0.0
    return float(valor) if pd.notna(valor) else 0.0

def mostrar_conexoes(grafo, no_inicial):
    #Mostra as conexoes de um cliente (campanhas e lojas) de forma hierarquica
    print(" -> Conexões diretas e indiretas:")

    visitados = {no_inicial}
    fila = deque([(no_inicial, 0)])

    while fila:
        no_atual, nivel = fila.popleft()

        #Limita a busca a 2 niveis para uma saida mais limpa
        if nivel > 2: continue

        if nivel > 0:
            print("    " * nivel + f"- {no_atual}")

        for vizinho in sorted(list(grafo.get(no_atual, []))):
            if vizinho not in visitados:
                visitados.add(vizinho)
                fila.append((vizinho, nivel + 1))

print("="*50)
print("="*50)
try:
    #Carregar os dados
    df_campanhas = pd.read_csv('CampaignQueue_semicolon.csv', sep=';')
    df_pedidos = pd.read_csv('Order_semicolon.csv', sep=';')

    #Limpa espacos nos nomes das colunas
    df_campanhas.columns = df_campanhas.columns.str.strip()
    df_pedidos.columns = df_pedidos.columns.str.strip()

    #Calcular a recorrência e os gastos de cada cliente
    recorrência = df_campanhas['customerId'].value_counts().reset_index()
    recorrência.columns = ['id_cliente', 'recorrência']

    df_pedidos['gasto'] = df_pedidos['totalAmount'].apply(limpar_valor)
    gastos = df_pedidos.groupby('customer')['gasto'].sum().reset_index()
    gastos.columns = ['id_cliente', 'gasto_total']

    #Juntar as informacoes em uma unica tabela
    metricas_clientes = pd.merge(recorrência, gastos, on='id_cliente', how='outer').fillna(0)

    #Encontrar os clientes mais recorrentes
    max_recorrência = metricas_clientes['recorrência'].max()
    clientes_top = metricas_clientes[metricas_clientes['recorrência'] == max_recorrência]

    cliente_menor_gasto = clientes_top.loc[clientes_top['gasto_total'].idxmin()]
    cliente_maior_gasto = clientes_top.loc[clientes_top['gasto_total'].idxmax()]

    #Preparar o grafo para encontrar as conexoes
    grafo_conexoes = {}
    for _, linha in df_campanhas.iterrows():
        cliente = f"Cliente_{linha['customerId']}"
        campanha = f"Campanha_{linha['campaignId']}"
        loja = f"Loja_{linha['storeId']}"

        for no in [cliente, campanha, loja]:
            if no not in grafo_conexoes: grafo_conexoes[no] = set()

        grafo_conexoes[cliente].add(campanha)
        grafo_conexoes[campanha].add(cliente)
        grafo_conexoes[campanha].add(loja)
        grafo_conexoes[loja].add(campanha)

    print("\n[ Cliente Mais Recorrente com MENOR Gasto ]")
    print(f" - ID: {int(cliente_menor_gasto['id_cliente'])}")

```

```
print(f" - Campanhas: {int(cliente_menor_gasto['recorrencia'])}")
print(f" - Gasto Total: R$ {cliente_menor_gasto['gasto_total']:.2f}")
mostrar_conexoes(grafo_conexoes, f"Cliente_{int(cliente_menor_gasto['id_cliente'])}")

print("\n" + "-"*50)

print("\n[ Cliente Mais Recorrente com MAIOR Gasto ]")
print(f" - ID: {int(cliente_maior_gasto['id_cliente'])}")
print(f" - Campanhas: {int(cliente_maior_gasto['recorrencia'])}")
print(f" - Gasto Total: R$ {cliente_maior_gasto['gasto_total']:.2f}")
mostrar_conexoes(grafo_conexoes, f"Cliente_{int(cliente_maior_gasto['id_cliente'])}")

except FileNotFoundError:
    print("\nERRO: Um dos arquivos CSV não foi encontrado. Verifique os nomes.")
except Exception as e:
    print(f"\nOcorreu um erro: {e}")

finally:
    print("\n" + "-"*50)
    print("="*50)
```