

```

def ler_arquivo():
    registros = []
    with open('registro.txt', 'r') as arquivo:
        next(arquivo) # Ignora a primeira linha
        for linha in arquivo:
            registros.append(linha.strip().split(','))
    return registros

def buscar_por_nome():
    registros = ler_arquivo()
    parte_nome = input("Digite parte do nome do cliente: ")
    clientes = set()
    for registro in registros:
        if parte_nome.lower() in registro[0].lower():
            clientes.add(registro[0])
    for cliente in clientes:
        print(cliente)

def buscar_por_cliente():
    registros = ler_arquivo()
    nome_cliente = input("Digite o nome completo do cliente: ")
    for registro in registros:
        if nome_cliente.lower() == registro[0].lower():
            print(f"Número do caso: {registro[1]}")

def informacoes_por_caso():
    registros = ler_arquivo()
    num_caso = input("Digite o número do caso: ")
    for registro in registros:
        if num_caso == registro[1]:
            nome = registro[0]
            despesa = float(registro[2])
            receita = float(registro[3])
            diferenca = receita - despesa
            print(f"Nome do cliente: {nome}")
            print(f"Despesa: {despesa}")
            print(f"Receita: {receita}")
            print(f"Diferença: {diferenca}")

def despesa_total():
    registros = ler_arquivo()
    total_despesa = sum(float(registro[2]) for registro in registros)
    print(f"Despesa Total: {total_despesa}")

def receita_total():
    registros = ler_arquivo()
    total_receita = sum(float(registro[3]) for registro in registros)
    print(f"Receita Total: {total_receita}")

def caso_maior_despesa():
    registros = ler_arquivo()
    maior_despesa = max(registros, key=lambda x: float(x[2]))
    print(f"Nome do cliente: {maior_despesa[0]}")
    print(f"Número do caso: {maior_despesa[1]}")
    print(f"Receita: {maior_despesa[3]}")
    print(f"Despesa: {maior_despesa[2]}")

def caso_maior_receita():
    registros = ler_arquivo()
    maior_receita = max(registros, key=lambda x: float(x[3]))

```

```

print(f"Nome do cliente: {maior_receita[0]}")
print(f"Número do caso: {maior_receita[1]}")
print(f"Receita: {maior_receita[3]}")
print(f"Despesa: {maior_receita[2]}")

def gravar_arquivo():
    registros = ler_arquivo()
    nome_cliente = input("Digite o nome completo do cliente: ")
    casos_cliente = []
    total_despesa = 0
    total_receita = 0
    for registro in registros:
        if nome_cliente.lower() == registro[0].lower():
            casos_cliente.append(registro)
            total_despesa += float(registro[2])
            total_receita += float(registro[3])
    with open(f'{nome_cliente}.txt', 'w') as arquivo:
        for caso in casos_cliente:
            arquivo.write(f"Nome do cliente: {caso[0]}\n")
            arquivo.write(f"Número do caso: {caso[1]}\n")
            arquivo.write(f"Despesa: {caso[2]}\n")
            arquivo.write(f"Receita: {caso[3]}\n\n")
        arquivo.write(f"Total de Despesas: {total_despesa}\n")
        arquivo.write(f"Total de Receitas: {total_receita}\n")
        arquivo.write(f"Diferença: {total_receita - total_despesa}\n")

def menu():
    opcoes = {
        'a': buscar_por_nome,
        'b': buscar_por_cliente,
        'c': informacoes_por_caso,
        'd': despesa_total,
        'e': receita_total,
        'f': caso_maior_despesa,
        'g': caso_maior_receita,
        'h': gravar_arquivo
    }
    while True:
        print("\nMenu:")
        print("(a) Buscar por parte do nome")
        print("(b) Buscar por nome completo do cliente")
        print("(c) Informações por número do caso")
        print("(d) Despesa total")
        print("(e) Receita total")
        print("(f) Caso com maior despesa")
        print("(g) Caso com maior receita")
        print("(h) Gravar arquivo por cliente")
        print("(q) Sair")
        opcao = input("Escolha uma opção: ").lower()
        if opcao == 'q':
            break
        elif opcao in opcoes:
            opcoes[opcao]()
        else:
            print("Opção inválida")

if __name__ == "__main__":
    menu()

```