

DAS - Documento de Arquitetura de Solução

Carga de Dados Financeiros

Autor: Fernando Alves da Costa

Sumário

<i>Resumo.....</i>	<i>3</i>
<i>1 – Escopo do Projeto</i>	<i>4</i>
<i>2 – Desenho de Arquitetura</i>	<i>4</i>
<i>3 – Hardware e Software utilizados</i>	<i>5</i>
<i>4 – Dados.....</i>	<i>6</i>
<i>5 - Detalhes da solução.....</i>	<i>6</i>
<i>6 – Conclusão</i>	<i>13</i>



Resumo

O presente documento apresenta uma descrição macro da implantação projeto Carga de Dados Financeiros. O objetivo desse documento é detalhar as ferramentas utilizadas.

1 – Escopo do Projeto

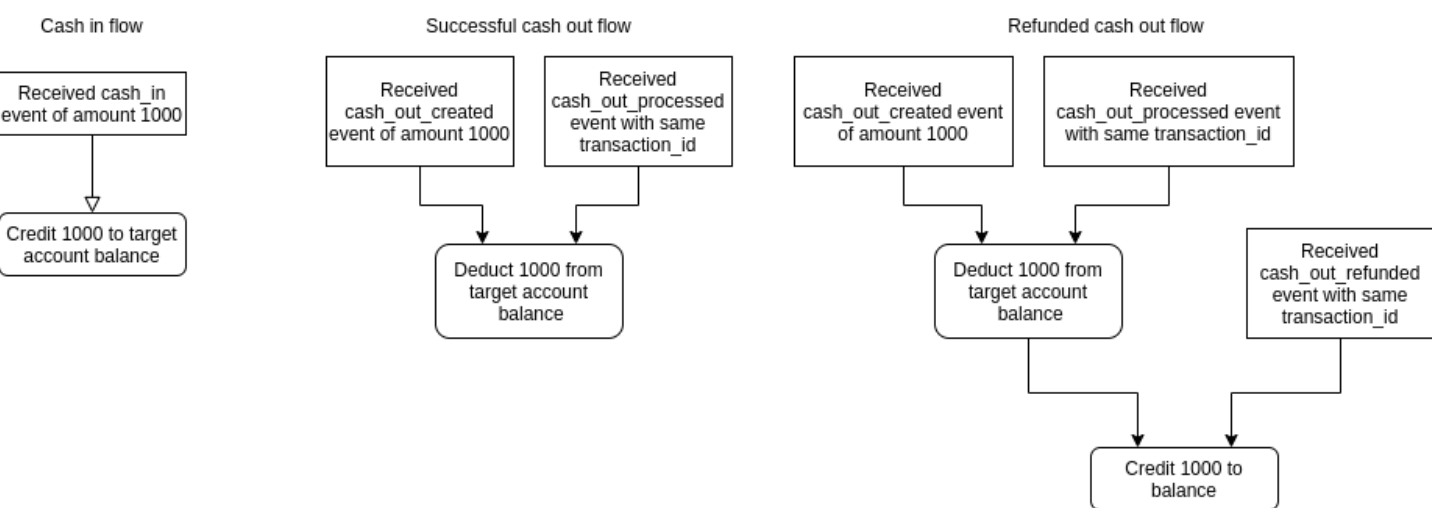
Dado um conjunto de tópicos do Kafka que descreve vários estados de transações financeiras, crie um aplicativo que processe esses eventos quase em tempo real e gere um saldo consolidado para um determinado par `account_id` e `datetime` solicitado.

- Os saldos devem:
 - incluem opções para filtrar por `account_id`, `datetime` e contraparte quando solicitado
 - mostrar todo o histórico de transações até o momento selecionado junto com o valor disponível na conta naquele momento.
- A aplicação deve ser preferencialmente escrita em Scala, Python ou Elixir.
- O reprocessamento dos eventos não deve levar a um estado que diverge da execução original.

O conjunto de dados é composto por 4 arquivos JSON, cada um dos quais deve ser processado como um tópico Kafka separado, como está. Etapas de processamento adicionais podem ser executadas posteriormente. Os estados de saque são representados como:

- criado: criado, mas ainda a ser processado, nenhum dinheiro deve sair da conta
- processado: a transação foi processada com sucesso, portanto, o saldo total deve ser deduzido do valor da transação correspondente
- reembolsado: a transação não foi concluída, o valor original deve retornar para a conta

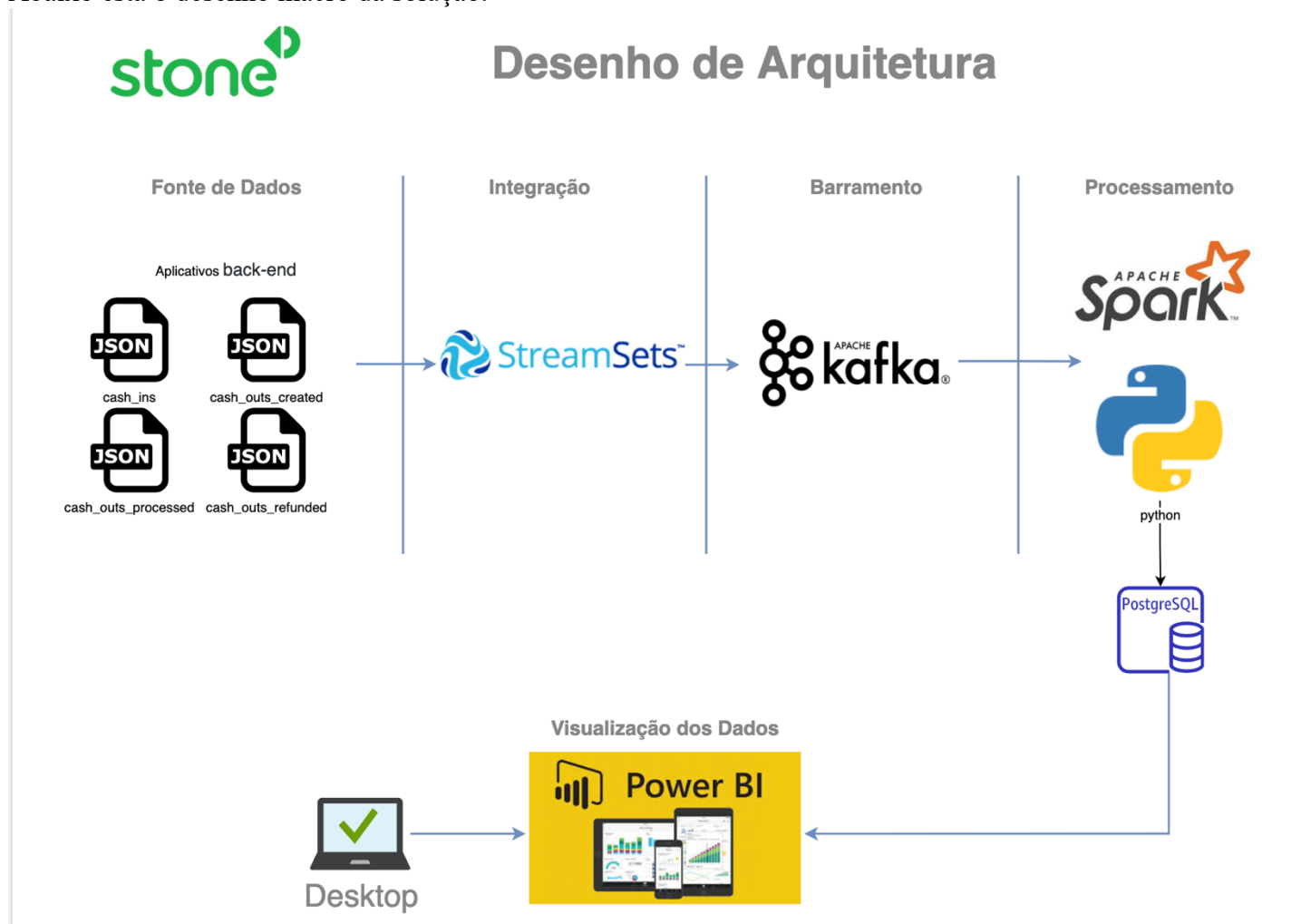
A figura a seguir ilustra os cenários possíveis:



2 – Desenho de Arquitetura

De acordo com a necessidade da área de negócio, foi definido uma arquitetura que atendesse por completo ao projeto.

Abaixo está o desenho macro da solução.



3 – Hardware e Software utilizados

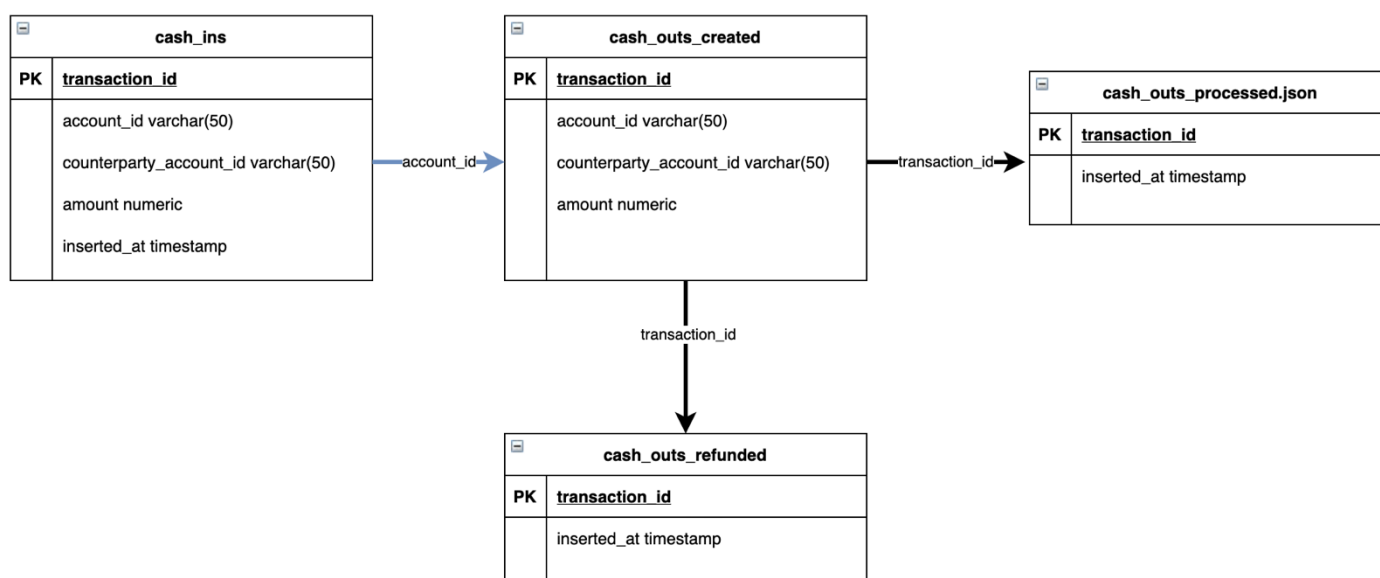
Ferramenta/Hardware	Função
Hardware Macbook i7, 1tb de Disco, 16gb de Memória	Hospedar o software de virtualização e as máquinas virtuais
Parallels Desktop	Software de virtualização de servidores
Máquina Virtual	Servidores Virtuais, compartilhando o mesmo hardware
CentOs	Sistema Operacional Linux, usando nas máquinas virtuais
Postgres	Banco de Dados relacional
StreamSets	Coletor de Dados, usado para fazer ingestão de dados
Kafka	plataforma de streaming de eventos distribuídos, usado em análise de streaming, integração de dados, etc

PowerBI Desktop	Ferramenta para visualização de dados, criação de Dashboard e relatórios.
Python	Linguagem de programação OpenSource
Apache Spark	Apache Spark é um mecanismo de análise unificado para processamento de dados em grande escala.

4 – Dados

- Foram disponibilizados pelo usuário quatro arquivos no formato json.

cash_outs_created.json
cash_outs_refunded.json
cash_outs_processed.json
cash_ins.json



5 - Detalhes da solução

- 1) Postgres – Criando um banco de dados para armazenar os quatros arquivos json, foi criada uma tabela para cada arquivo.

```
CREATE TABLE cash_ins (
  transaction_id varchar(50) primary key,
  account_id varchar(50),
  counterparty_account_id varchar(50),
  amount numeric,
  inserted_at timestamp);
```

```
CREATE TABLE cash_outs_created (
  transaction_id varchar(50) primary key,
```

```

account_id varchar(50),
counterparty_account_id varchar(50),
amount numeric,
inserted_at timestamp)

CREATE TABLE cash_outs_processed (
transaction_id varchar(50) primary key,
inserted_at timestamp)

CREATE TABLE cash_outs_refunded (
transaction_id varchar(50) primary key,
inserted_at timestamp)

```

- 2) Kafka – Criado quatro tópicos no Kafka, cada tópico para armazenar seu arquivo correspondente.

```

/opt/kafka/bin/kafka-topics.sh --create --topic cash_ins --zookeeper localhost:2181 --replication-factor 1 --partitions 1
/opt/kafka/bin/kafka-topics.sh --create --topic cash_outs_created --zookeeper localhost:2181 --replication-factor 1 --partitions 1
/opt/kafka/bin/kafka-topics.sh --create --topic cash_outs_processed --zookeeper localhost:2181 --replication-factor 1 --partitions 1
/opt/kafka/bin/kafka-topics.sh --create --topic cash_outs_refunded --zookeeper localhost:2181 --replication-factor 1 --partitions 1

```

- 3) StreamSets – Criado quatro coletores, um para cada arquivo json. O coletor irá monitorar o arquivo json, caso um novo registro seja inserido, o mesmo será enviado para o tópico Kafka.
- 4) Python – Criado um programa que fica monitorando os tópicos no Kafka, caso chegue um novo registro, o mesmo é inserido no banco de dados em tempo real. Para a execução do programa, passar o nome do tópico por parâmetro.

```

# Programa para carga de topico no kafka
# Autor: Fernando Alves da Costa
# Data Criação: 14/01/2021

# Bibliotecas utilizadas

from kafka import KafkaConsumer
from json import loads
import psycog2
import ast
import sys
import datetime

# Tratando os parametros de entrada
param1=[]
param1=str(' '.join(sys.argv[1:])).split()

if param1 == "":
    print("Favor, passar o parametro com o nome do tópico")
    sys.exit()
else:
    print("Parametro encontrado")

```

```
# Parametros de banco de dados
host = "postgres02"
dbname = "dw"
user = "dw"
password = "Dw01"
port = "5432"

# inicio do loop, o programa foi feito para passar varios tópicos como entrada
for param in param1:
    print("Processando o topico: ",param)

    # Conexao com o banco de dados
    conn_string = "host={0} user={1} dbname={2} password={3} port={4}".format(host, user, dbname, password,
port)
    conn = psycopg2.connect(conn_string)
    try:
        print("Sucesso ao conectar no banco de dados")
    except:
        print("Erro ao conectar no banco de dados")
        sys.exit()

    # Conexao com o kafka com método consumer
    # Para consumir o tópico desde o inicio, alterar o parametro auto_offset_reset='earliest'
    # Para nao deixar o programa rodando sem parar, incluir o parametro consumer_timeout_ms = 15000
    consumer = KafkaConsumer(param, bootstrap_servers='kafka01:9092', consumer_timeout_ms = 15000,
auto_offset_reset='latest', enable_auto_commit=True)
    try:
        print("Sucesso ao conectar no kafka")
    except:
        print("Erro ao conectar no kafka")
        sys.exit()

    # Verifica qual o tópico, para definir tabela no postgres

def verifica_layout_arquivo():
    global transaction_id_
    global account_id_
    global counterparty_account_id_
    global amount_
    global inserted_at_
    global postgres_insert_query
    global record_to_insert

    # Preenchendo as variaveis para carga no postgres

    if param == 'cash_ins':
        transaction_id_ = mydata.get('transaction_id')
        account_id_ = mydata.get('account_id')
        counterparty_account_id_ = mydata.get('counterparty_account_id')
        amount_ = mydata.get('amount')
        inserted_at_ = datetime.datetime.fromtimestamp(mydata.get('inserted_at') / 1e3)
        postgres_insert_query = """ insert into cash_ins (transaction_id, account_id, counterparty_account_id,
amount, inserted_at) values (%s, %s, %s, %s, %s)"""
        record_to_insert = (transaction_id_, account_id_, counterparty_account_id_, amount_, inserted_at_)
    elif param == 'cash_outs_created':
```



```

transaction_id_=mydata.get('transaction_id')
account_id_=mydata.get('account_id')
counterparty_account_id_=mydata.get('counterparty_account_id')
amount_=mydata.get('amount')
inserted_at_=datetime.datetime.fromtimestamp(mydata.get('inserted_at') / 1e3)
postgres_insert_query = """ insert into cash_outs_created (transaction_id, account_id,
counterparty_account_id, amount, inserted_at) values (%s, %s, %s, %s, %s)"""
record_to_insert = (transaction_id_, account_id_, counterparty_account_id_, amount_, inserted_at_)
elif param == 'cash_outs_refunded':
    transaction_id_=mydata.get('transaction_id')
    inserted_at_=datetime.datetime.fromtimestamp(mydata.get('inserted_at') / 1e3)
    postgres_insert_query = """ insert into cash_outs_refunded (transaction_id, inserted_at) values (%s, %s)"""
    record_to_insert = (transaction_id_, inserted_at_)
elif param == 'cash_outs_processed':
    transaction_id_=mydata.get('transaction_id')
    inserted_at_=datetime.datetime.fromtimestamp(mydata.get('inserted_at') / 1e3)
    postgres_insert_query = """ insert into cash_outs_processed (transaction_id, inserted_at) values (%s, %s)"""
    record_to_insert = (transaction_id_, inserted_at_)
else:
    print("Verificar parametro")
    sys.exit()

contador=0

# leitura do conteudo extraido do kafka
for message in consumer:
    linha=message.value
    dict_str = linha.decode("utf-8")
    mydata = ast.literal_eval(dict_str)

# Carregando os registros no postgres
verifica_layout_arquivo()
cur = conn.cursor()
cur.execute(postgres_insert_query, record_to_insert)
cur.close()
conn.commit()

contador = contador + 1

print("Registros carregados: ",contador)

consumer.close()
conn.close()

```

- 5) **Apache Spark** – Para o processamento em escala, foi implementado um cluster Spark, que executa o programa python, assim o processamento será distribuído no cluster.

```

spark-submit --master yarn consumer_final.py cash_outs_created
spark-submit --master yarn consumer_final.py cash_outs_processed
spark-submit --master yarn consumer_final.py cash_outs_refunded
spark-submit --master yarn consumer_final.py cash_ins

```

6) PowerBI – Criado um dashboard e um relatório no PowerBI, com opções de filtros diversos.

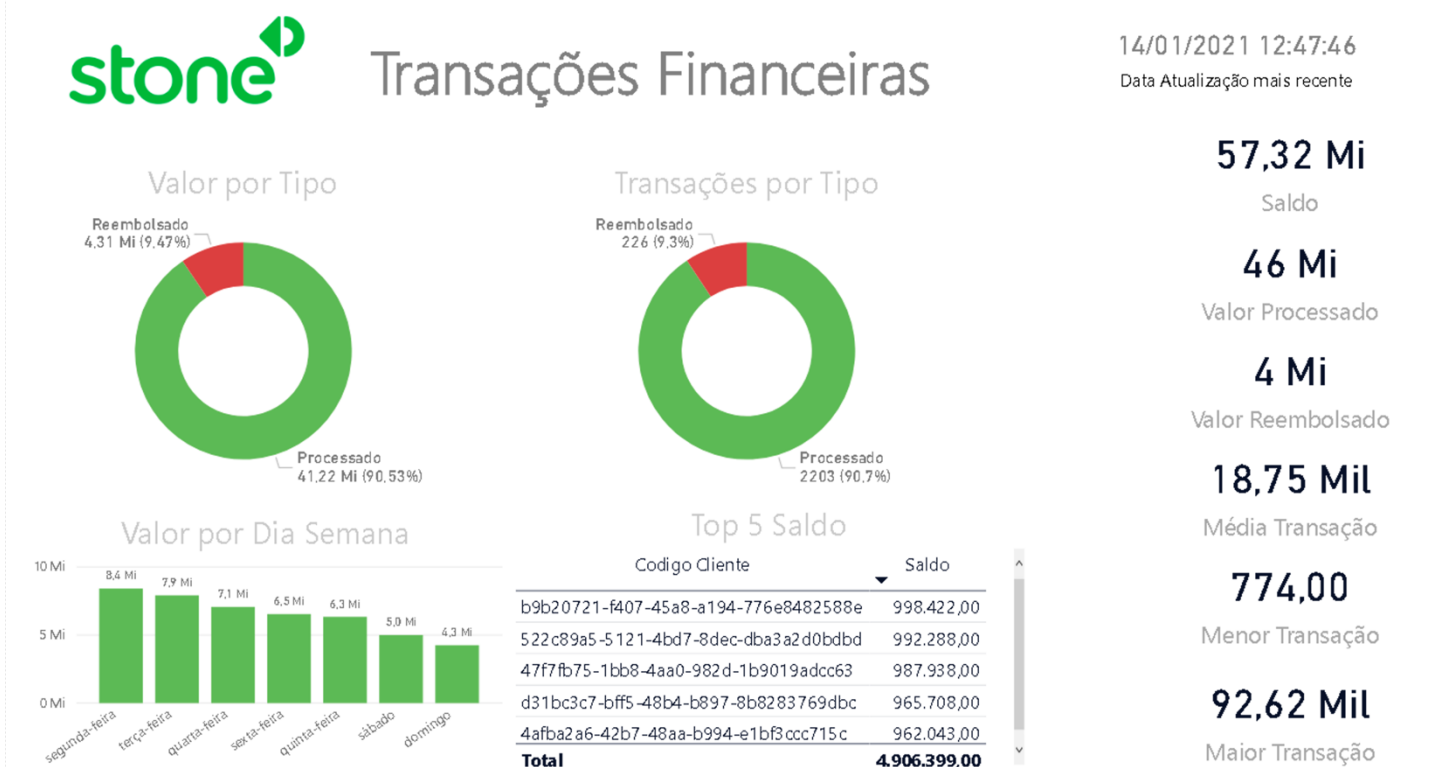


Figura 1 - DashBoard mostrando uma visão geral

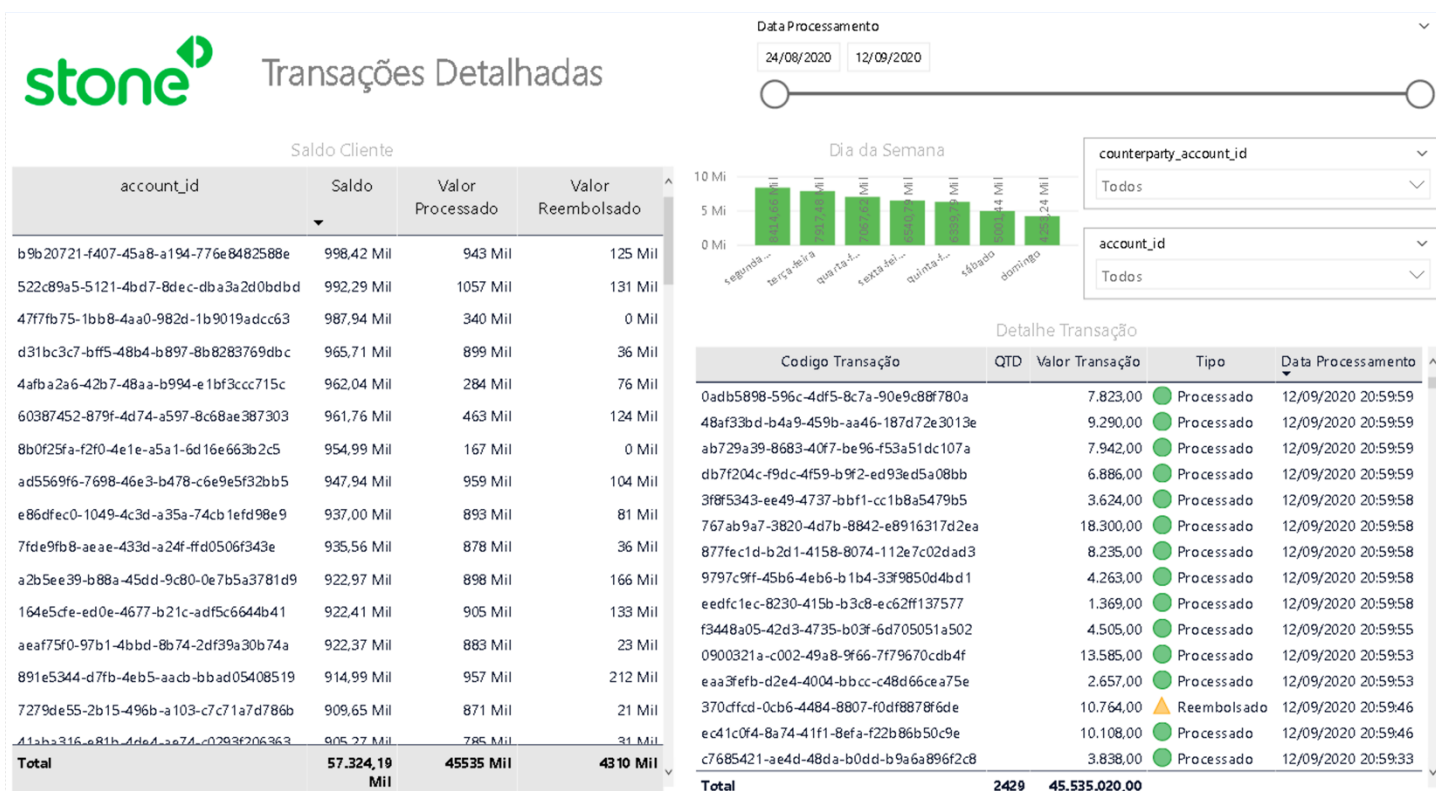


Figura 2 - Relatório detalhado, com diversas opções de filtros

Transações Detalhadas

Saldo Cliente

account_id	Saldo	Valor Processado	Valor Reembolsado
b9b20721-f407-45a8-a194-776e8482588e	998,42 Mil	943 Mil	125 Mil
522c89a5-5121-4bd7-8dec-dba3a2d0bdbc	992,29 Mil	1057 Mil	131 Mil
47f7fb75-1bb8-4aa0-982d-1b9019adcc63	987,94 Mil	340 Mil	0 Mil
d31bc3c7-bff5-48b4-b897-8b8283769dbc	965,71 Mil	899 Mil	36 Mil
4afb2a6-42b7-48aa-b994-e1bf3ccc715c	962,04 Mil	284 Mil	76 Mil
60387452-879f-4d74-a597-8c68ae387303	961,76 Mil	463 Mil	124 Mil
8b0f25fa-f2f0-4e1e-a5a1-6d16e663b2c5	954,99 Mil	167 Mil	0 Mil
ad5569f6-7698-46e3-b478-c6e9e5f32bb5	947,94 Mil	959 Mil	104 Mil
e86dfec0-1049-4c3d-a35a-74cb1efd98e9	937,00 Mil	893 Mil	81 Mil
7fde9fb8-aeae-433d-a24f-ffd0506f343e	935,56 Mil	878 Mil	36 Mil
a2b5ee39-b88a-45dd-9c80-0e7b5a3781d9	922,97 Mil	898 Mil	166 Mil
164e5cfe-ed0e-4677-b21c-adf5c6644b41	922,41 Mil	905 Mil	133 Mil
aeaf75f0-97b1-4bbd-8b74-2df39a30b74a	922,37 Mil	883 Mil	23 Mil
891e5344-d7fb-4eb5-aacb-bbad05408519	914,99 Mil	957 Mil	212 Mil
7279de55-2b15-496b-a103-c7c71a7d786b	909,65 Mil	871 Mil	21 Mil
41ab316-a81b-4de4-a74c-c0293f206363	905,27 Mil	785 Mil	31 Mil
Total	57.324,19 Mil	45535 Mil	4310 Mil

Data Processamento

24/08/2020 12/09/2020



Dia da Semana



counterparty_account_id

Todos

account_id

Todos

Detalhe Transação

Codigo Transação	QTD	Valor Transação	Tipo	Data Processamento
8059ecb9-5882-4be9-9509-59ac7c6ad60b		56.574,00	Processado	10/09/2020 20:59:57
fa8a7ec1-b1df-4c95-a2b9-1a4fbc2f95ae		42.430,00	Processado	10/09/2020 20:59:52
ee074348-389d-4384-9e84-68efbbb0f653		42.430,00	Processado	10/09/2020 20:59:32
10c9f4d7-8d9e-4d59-81b7-b5e1d6f70567		84.861,00	Processado	10/09/2020 20:55:34
91ef0926-fa62-4f66-932e-0e811dda4532		49.918,00	Reembolsado	10/09/2020 19:50:36
bea8554-0dac-4b6d-a6cd-4a616152eb9f		56.574,00	Processado	02/09/2020 20:59:45
83e47dcd-0c21-4846-ad28-39c403ca2235		56.574,00	Processado	02/09/2020 20:40:17
79647eab-2842-46ef-b355-a781b9c97879		73.981,00	Reembolsado	02/09/2020 18:00:30
Total	8	463.342,00		

Figura 3 - Na tabela a esquerda, com informações do saldo, valor processado e valor reembolsado por cliente, basta escolher um cliente, todos os dados são filtrados para esse cliente. Do lado direito, é mostrado o detalhamento das transações do cliente selecionado.

Transações Detalhadas

Saldo Cliente

account_id	Saldo	Valor Processado	Valor Reembolsado
b9b20721-f407-45a8-a194-776e8482588e	998,42 Mil	943 Mil	125 Mil
522c89a5-5121-4bd7-8dec-dba3a2d0bdbc	992,29 Mil	1057 Mil	131 Mil
47f7fb75-1bb8-4aa0-982d-1b9019adcc63	987,94 Mil	340 Mil	0 Mil
d31bc3c7-bff5-48b4-b897-8b8283769dbc	965,71 Mil	899 Mil	36 Mil
4afb2a6-42b7-48aa-b994-e1bf3ccc715c	962,04 Mil	284 Mil	76 Mil
60387452-879f-4d74-a597-8c68ae387303	961,76 Mil	463 Mil	124 Mil
8b0f25fa-f2f0-4e1e-a5a1-6d16e663b2c5	954,99 Mil	167 Mil	0 Mil
ad5569f6-7698-46e3-b478-c6e9e5f32bb5	947,94 Mil	959 Mil	104 Mil
e86dfec0-1049-4c3d-a35a-74cb1efd98e9	937,00 Mil	893 Mil	81 Mil
7fde9fb8-aeae-433d-a24f-ffd0506f343e	935,56 Mil	878 Mil	36 Mil
a2b5ee39-b88a-45dd-9c80-0e7b5a3781d9	922,97 Mil	898 Mil	166 Mil
164e5cfe-ed0e-4677-b21c-adf5c6644b41	922,41 Mil	905 Mil	133 Mil
aeaf75f0-97b1-4bbd-8b74-2df39a30b74a	922,37 Mil	883 Mil	23 Mil
891e5344-d7fb-4eb5-aacb-bbad05408519	914,99 Mil	957 Mil	212 Mil
7279de55-2b15-496b-a103-c7c71a7d786b	909,65 Mil	871 Mil	21 Mil
41ab316-a81b-4de4-a74c-c0293f206363	905,27 Mil	785 Mil	31 Mil
Total	57.324,19 Mil	45535 Mil	4310 Mil

Data Processamento

24/08/2020 12/09/2020



setembro 2020

dom	seg	ter	qua	qui	sex	sáb
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

Detalhe Transação

Codigo Transação	QTD	Valor Transação	Tipo	Data Processamento
8059ecb9-5882-4be9-9509-59ac7c6ad60b		56.574,00	Processado	10/09/2020 20:59:57
fa8a7ec1-b1df-4c95-a2b9-1a4fbc2f95ae		42.430,00	Processado	10/09/2020 20:59:52
ee074348-389d-4384-9e84-68efbbb0f653		42.430,00	Processado	10/09/2020 20:59:32
10c9f4d7-8d9e-4d59-81b7-b5e1d6f70567		84.861,00	Processado	10/09/2020 20:55:34
91ef0926-fa62-4f66-932e-0e811dda4532		49.918,00	Reembolsado	10/09/2020 19:50:36
bea8554-0dac-4b6d-a6cd-4a616152eb9f		56.574,00	Processado	02/09/2020 20:59:45
83e47dcd-0c21-4846-ad28-39c403ca2235		56.574,00	Processado	02/09/2020 20:40:17
79647eab-2842-46ef-b355-a781b9c97879		73.981,00	Reembolsado	02/09/2020 18:00:30
Total	8	463.342,00		

Figura 4 - Opção de filtro por data, pode ser uma data específica ou um intervalo de datas.

Saldo Cliente

account_id	Saldo	Valor Processado	Valor Reembolsado
b9b20721-f407-45a8-a194-776e8482588e	998,42 Mil	943 Mil	125 Mil
522c89a5-5121-4bd7-8dec-dba3a2d0bdbc	992,29 Mil	1057 Mil	131 Mil
47f7fb75-1bb8-4aa0-982d-1b9019adcc63	987,94 Mil	340 Mil	0 Mil
d31bc3c7-bff5-48b4-b897-8b8283769dbc	965,71 Mil	899 Mil	36 Mil
4afb2a6-42b7-48aa-b994-e1bf3ccc715c	962,04 Mil	284 Mil	76 Mil
60387452-879f-4d74-a597-8c68ae387303	961,76 Mil	463 Mil	124 Mil
8b0f25fa-f2f0-4e1e-a5a1-6d16e663b2c5	954,99 Mil	167 Mil	0 Mil
ad5569f6-7698-46e3-b478-c6e9e5f32bb5	947,94 Mil	959 Mil	104 Mil
e86dfec0-1049-4c3d-a35a-74cb1efd98e9	937,00 Mil	893 Mil	81 Mil
7fde9fb8-aeae-433d-a24f-ffd0506f343e	935,56 Mil	878 Mil	36 Mil
a2b5ee39-b88a-45d4-9c80-0e7b5a3781d9	922,97 Mil	898 Mil	166 Mil
164e5cfe-ed0e-4677-b21c-adf5c6644b41	922,41 Mil	905 Mil	133 Mil
aeaf75f0-97b1-4bbd-8b74-2df39a30b74a	922,37 Mil	883 Mil	23 Mil
891e5344-d7fb-4eb5-aacb-bbad05408519	914,99 Mil	957 Mil	212 Mil
7279de55-2b15-496b-a103-c7c71a7d786b	909,65 Mil	871 Mil	21 Mil
41ba316-e81b-4de4-a74c-c0293f206363	905,27 Mil	785 Mil	31 Mil
Total	57.324,19 Mil	45535 Mil	4310 Mil

Data Processamento

24/08/2020 12/09/2020



Detalhe Tran

Codigo Transação	QTD	Valor Tra	counterparty_account_id	Status	Data
0adb5898-596c-4df5-8c7a-90e9c88f780a	7	7.942,00	00883ba6-e8c4-454c-9274-719dbe70a38	Processado	12/09/2020 20:59:59
48af33bd-b4a9-459b-aa46-187d72e3013e	9	6.886,00	066998f6-21d8-48bc-82f7-37a1f71a5e2a	Processado	12/09/2020 20:59:59
ab729a39-8683-40f7-be96-f53a51dc107a	7	3.624,00	0733e16b-a25d-4a91-b840-ab847359e3d8	Processado	12/09/2020 20:59:58
db7f204c-f9dc-4f59-b9f2-ed93ed5a08bb	9	18.300,00	07d1dd4a-2006-4717-8a4c-387e1c8a2026	Processado	12/09/2020 20:59:58
3f8f5343-ee49-4737-bbf1-cc1b8a5479b5	7	8.235,00	0d3793a2-410d-4fe9-9755-857a6c345721	Processado	12/09/2020 20:59:58
767ab9a7-3820-4d7b-8842-e8916317d2ea	7	4.263,00	119b859c-0ca9-4fa1-b0ce-7f5612f24e0a	Processado	12/09/2020 20:59:58
877fec1d-b2d1-4158-8074-112e7c02dad3	7	1.369,00	1240c29f-a49f-4dbe-99aa-0a6121a6c6fd	Processado	12/09/2020 20:59:55
9797c9ff-45b6-4eb6-b1b4-33f9850d4bd1	6	4.505,00	13f94ed7-1d65-4b95-a22f-a2295a2e88d6	Processado	12/09/2020 20:59:53
eedfc1ec-8230-415b-b3c8-e62ff137577	3	2.657,00	1652c4ba-568f-45c7-80de-7fc6d14d0d6a	Processado	12/09/2020 20:59:53
f3448a05-42d3-4735-b03f-6d705051a502	6	10.764,00		Reembolsado	12/09/2020 20:59:46
0900321a-c002-49a8-9f66-7f79670cdb4f	6	10.108,00		Processado	12/09/2020 20:59:46
eaaf3efb-d2e4-4004-bbcc-c48d66ce7a75e	3	3.838,00		Processado	12/09/2020 20:59:33
370dfcd-0cb6-4484-8807-f0df8878f6de					
ec41c0f4-8a74-41f1-8efa-f22b86b50c9e					
c7685421-ae4d-48da-b0dd-b9a6a896f2c8					
Total	2429	45.535.020,00			

Figura 5 - Opção de filtro por counterparty.

Saldo Cliente

account_id	Saldo	Valor Processado	Valor Reembolsado
b9b20721-f407-45a8-a194-776e8482588e	998,42 Mil	943 Mil	125 Mil
522c89a5-5121-4bd7-8dec-dba3a2d0bdbc	992,29 Mil	1057 Mil	131 Mil
47f7fb75-1bb8-4aa0-982d-1b9019adcc63	987,94 Mil	340 Mil	0 Mil
d31bc3c7-bff5-48b4-b897-8b8283769dbc	965,71 Mil	899 Mil	36 Mil
4afb2a6-42b7-48aa-b994-e1bf3ccc715c	962,04 Mil	284 Mil	76 Mil
60387452-879f-4d74-a597-8c68ae387303	961,76 Mil	463 Mil	124 Mil
8b0f25fa-f2f0-4e1e-a5a1-6d16e663b2c5	954,99 Mil	167 Mil	0 Mil
ad5569f6-7698-46e3-b478-c6e9e5f32bb5	947,94 Mil	959 Mil	104 Mil
e86dfec0-1049-4c3d-a35a-74cb1efd98e9	937,00 Mil	893 Mil	81 Mil
7fde9fb8-aeae-433d-a24f-ffd0506f343e	935,56 Mil	878 Mil	36 Mil
a2b5ee39-b88a-45d4-9c80-0e7b5a3781d9	922,97 Mil	898 Mil	166 Mil
164e5cfe-ed0e-4677-b21c-adf5c6644b41	922,41 Mil	905 Mil	133 Mil
aeaf75f0-97b1-4bbd-8b74-2df39a30b74a	922,37 Mil	883 Mil	23 Mil
891e5344-d7fb-4eb5-aacb-bbad05408519	914,99 Mil	957 Mil	212 Mil
7279de55-2b15-496b-a103-c7c71a7d786b	909,65 Mil	871 Mil	21 Mil
41ba316-e81b-4de4-a74c-c0293f206363	905,27 Mil	785 Mil	31 Mil
Total	57.324,19 Mil	45535 Mil	4310 Mil

Data Processamento

24/08/2020 12/09/2020



Detalhe Tran

Codigo Transação	QTD	Valor Tra	account_id	Status	Data
0adb5898-596c-4df5-8c7a-90e9c88f780a	7	7.942,00	02d75fa6-9919-4a10-a4a4-d214f5715eae	Processado	12/09/2020 20:59:58
48af33bd-b4a9-459b-aa46-187d72e3013e	9	6.886,00	040e9641-5b6c-4369-acb4-168df29c38de	Processado	12/09/2020 20:59:58
ab729a39-8683-40f7-be96-f53a51dc107a	7	3.624,00	0686d71a-eb96-4f4d-b592-e4bd802a17cb	Processado	12/09/2020 20:59:58
db7f204c-f9dc-4f59-b9f2-ed93ed5a08bb	9	18.300,00	0b00b76d-999d-4114-8d97-f13f597c402f	Processado	12/09/2020 20:59:58
3f8f5343-ee49-4737-bbf1-cc1b8a5479b5	7	8.235,00	0b7d260d-e477-4fca-ad62-02b89366b4cf	Processado	12/09/2020 20:59:58
767ab9a7-3820-4d7b-8842-e8916317d2ea	7	4.263,00	0bd1db2c-3f91-49b5-a3bb-c77ab03ba4c7	Processado	12/09/2020 20:59:58
877fec1d-b2d1-4158-8074-112e7c02dad3	7	1.369,00	15c35118-95bf-46d0-bc3f-a2f02b73f0cc	Processado	12/09/2020 20:59:58
9797c9ff-45b6-4eb6-b1b4-33f9850d4bd1	6	4.505,00	164e5cfe-ed0e-4677-b21c-adf5c6644b41	Processado	12/09/2020 20:59:55
eedfc1ec-8230-415b-b3c8-e62ff137577	3	2.657,00	17785f8e-feb1-4025-8cde-3bed6a352fe0	Processado	12/09/2020 20:59:53
f3448a05-42d3-4735-b03f-6d705051a502	6	10.764,00		Reembolsado	12/09/2020 20:59:46
0900321a-c002-49a8-9f66-7f79670cdb4f	6	10.108,00		Processado	12/09/2020 20:59:46
eaaf3efb-d2e4-4004-bbcc-c48d66ce7a75e	3	3.838,00		Processado	12/09/2020 20:59:33
370dfcd-0cb6-4484-8807-f0df8878f6de					
ec41c0f4-8a74-41f1-8efa-f22b86b50c9e					
c7685421-ae4d-48da-b0dd-b9a6a896f2c8					
Total	2429	45.535.020,00			

Figura 6 - Opção de filtro por account_id.

6 – Conclusão

Documento feito com o propósito de dar uma noção macro do funcionamento, de como é feito o processamento e como foi utilizado cada ferramenta.