# Apache Kafka

Anderson Giuseppe Saraiva Patriarca Fantin – 9016981
Gustavo Câmara Reis – 11207780
Laís Garcia – 11270628
Larissa Fabião da Fonseca – 11208367
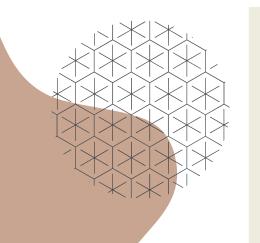
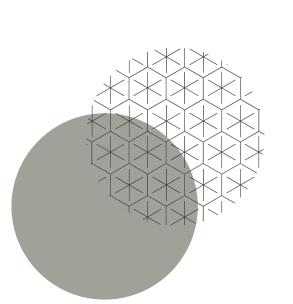# Agenda

# OI

# O que é o Apache Kafka?

# O que é o Apache Kafka

## Plataforma open-source distribuída de streaming
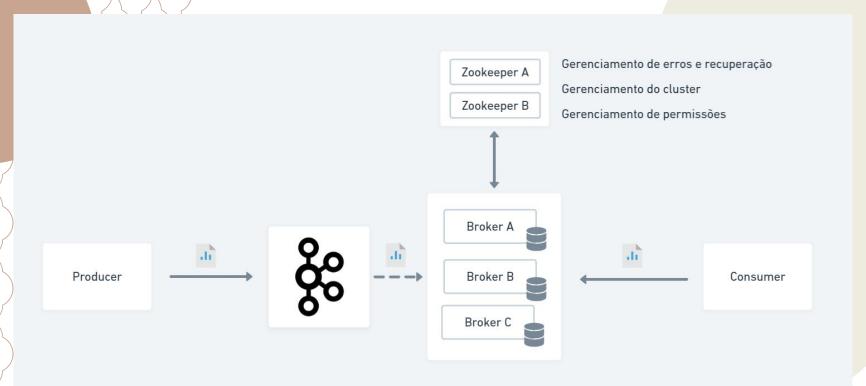
- Scala e Java
- Objetivo
- Fluxo básico com mensagens
- Qual a utilidade?
- Proporciona diversas maneiras de mover dados
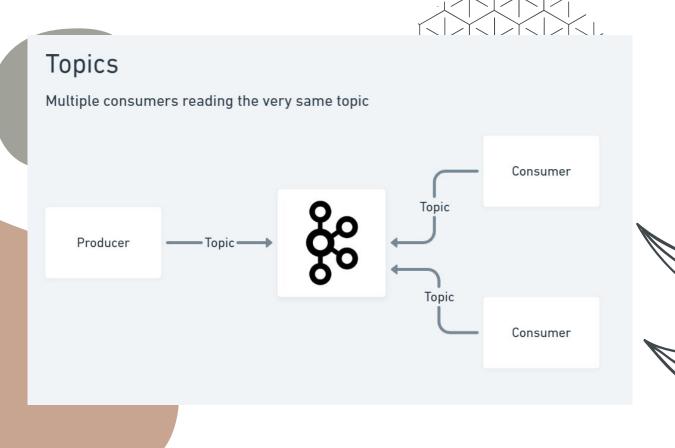
# 02
# Como funciona?

# Ciclo geral
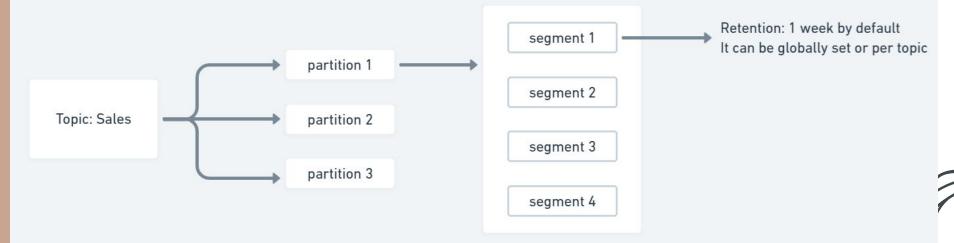
# Armazenamento de informações
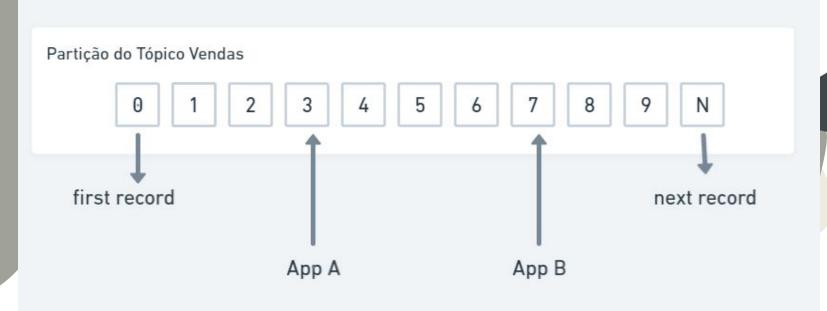
## Topics

Multiple consumers reading the very same topic

# Armazenamento de informações



Topic partitions and segments

Topic: Sales → partition 1 → segment 1 → Retention: 1 week by default. It can be globally set or per topic

partition 2

partition 3

segment 2

segment 3

segment 4

# Leitura de informações



## Topic log

Partição do Tópico Vendas

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | N |

first record

App A

App B

next record

# Leitura de informações

## Record anatomy

| | |
|---|---|
| **Record 1** → | Headers → Optional |
| | Key → Logical information context. Ex: Shipping |
| | Value → Payload |
| | Timestamp |

# Leitura de informações
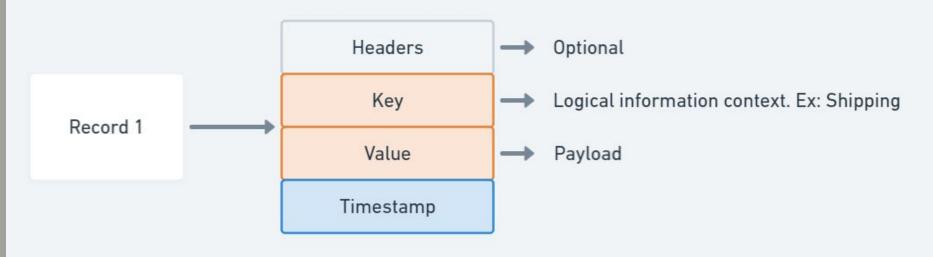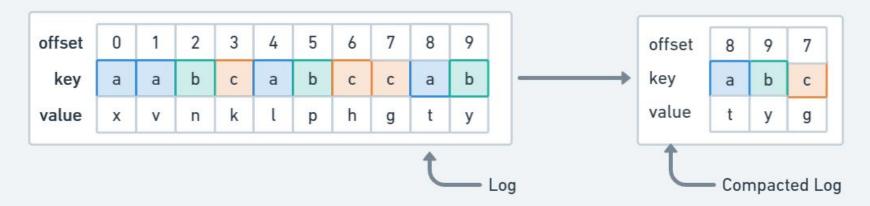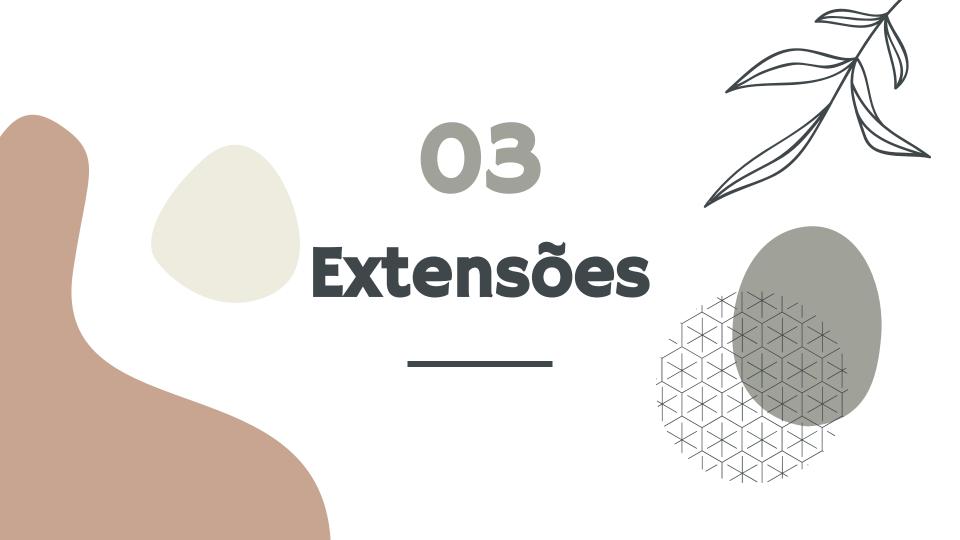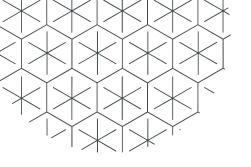
## Compacted topic

Topic: Sales

| offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|---|
| key | a | a | b | c | a | b | c | c | a | b |
| value | x | v | n | k | l | p | h | g | t | y |

Log

| offset | 8 | 9 | 7 |
|--------|---|---|---|
| key | a | b | c |
| value | t | y | g |

Compacted Log

# 03

# Extensões

———

# Extensões

## Connect
Processa e analiza dados armazenados no Kafka
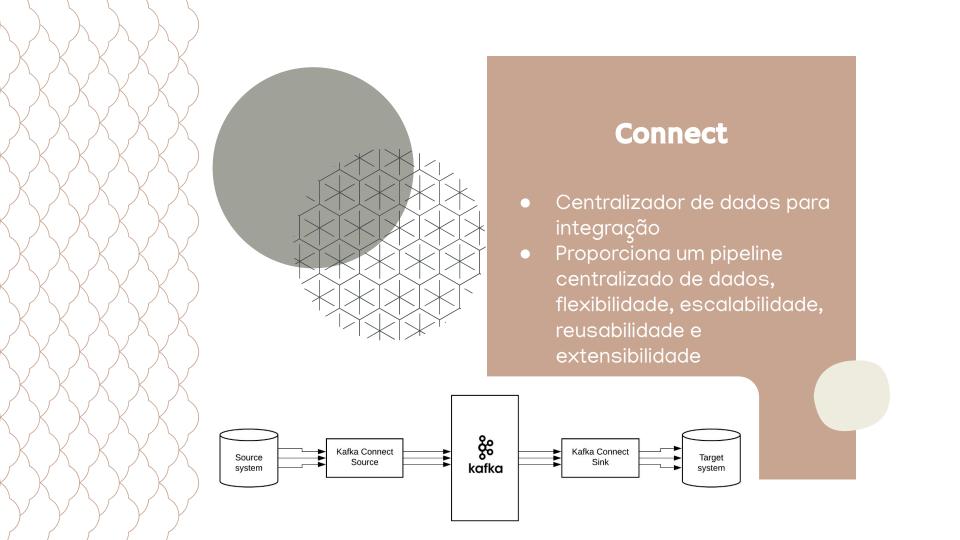
## Streams
Framework para tratar, transformar e agregar as mensagens
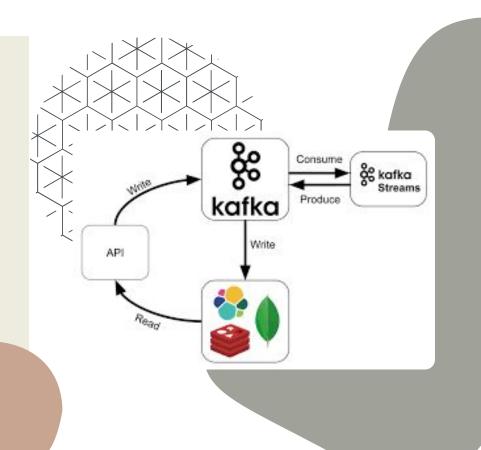
## Schema Registry
Biblioteca de cliente onde os dados são armazenados em um cluster Kafka

# Connect

- Centralizador de dados para integração
- Proporciona um pipeline centralizado de dados, flexibilidade, escalabilidade, reusabilidade e extensibilidade

| Source system | → | Kafka Connect Source | → | kafka | → | Kafka Connect Sink | → | Target system |
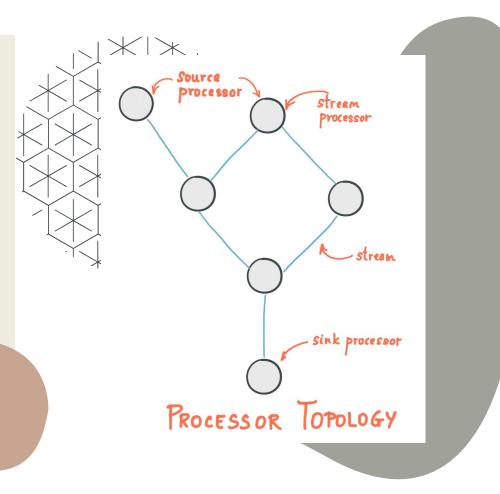
# Streams
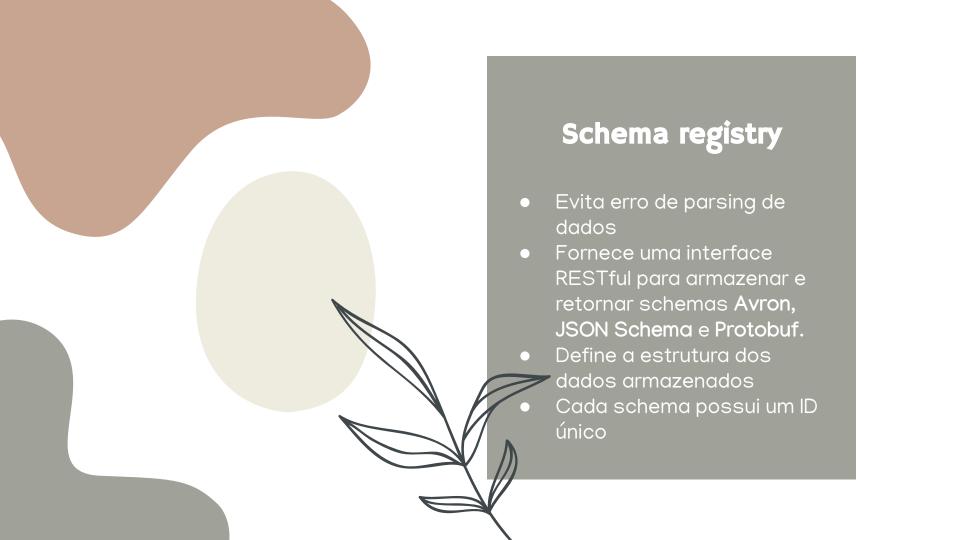
- API Funcional de Java
- Permite a criação de filtros, agregações, junções entre outras operações
- Escalável
- Tolerante a falhas
- Se integra como uma biblioteca
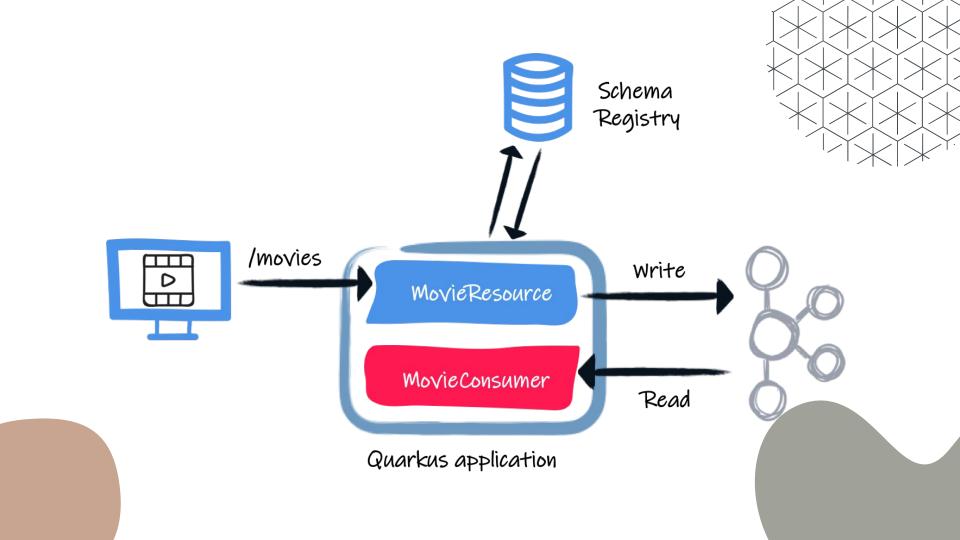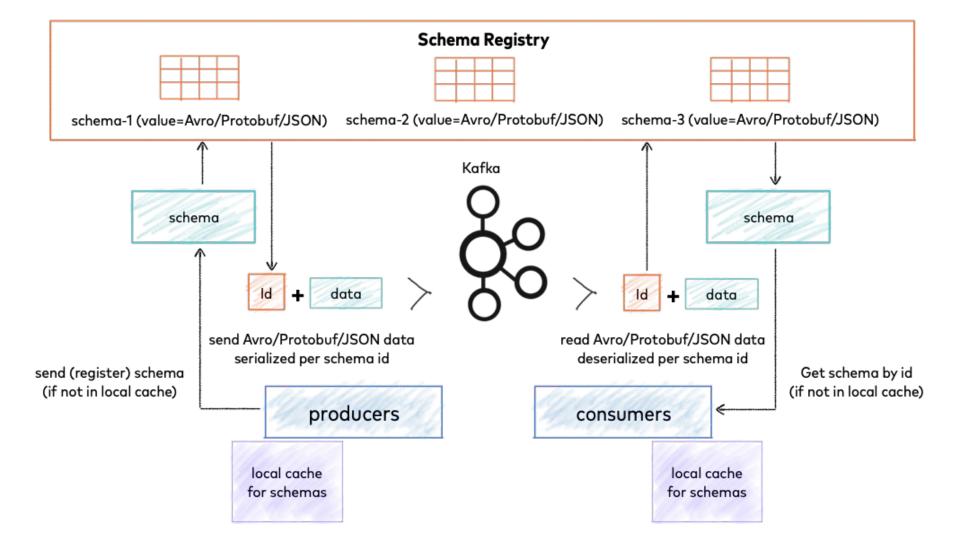
# Streams
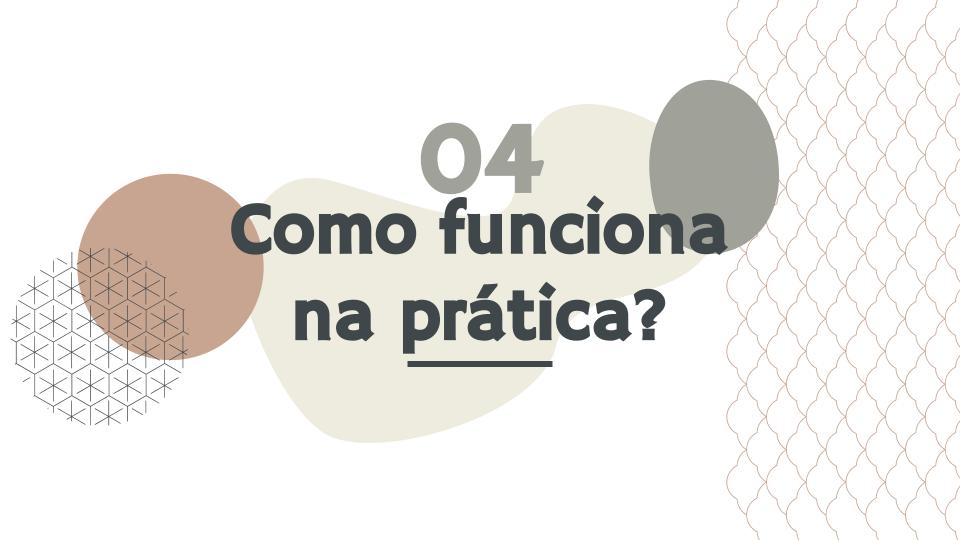
- **Stream:** uma sequência ordenada, reproduzível e tolerante a falhas de registros de dados
- **Stream processor:** um step de processamento
- **Source processor:** produz um fluxo de entrada consumindo registros do Kafka
- **Sink processor:** envia os registros recebidos de seus processadores up-stream para um tópico Kafka.

Source processor

Stream processor

stream

sink processor

PROCESSOR TOPOLOGY

# Schema registry

- Evita erro de parsing de dados
- Fornece uma interface RESTful para armazenar e retornar schemas **Avron, JSON Schema** e **Protobuf.**
- Define a estrutura dos dados armazenados
- Cada schema possui um ID único

Schema Registry

/movies

MovieResource

MovieConsumer

Write

Read

Quarkus application

**Schema Registry**

schema-1 (value=Avro/Protobuf/JSON)   schema-2 (value=Avro/Protobuf/JSON)   schema-3 (value=Avro/Protobuf/JSON)

schema

schema

Kafka

Id + data

Id + data

send Avro/Protobuf/JSON data
serialized per schema id

read Avro/Protobuf/JSON data
deserialized per schema id

send (register) schema
(if not in local cache)

Get schema by id
(if not in local cache)

producers

consumers

local cache
for schemas

local cache
for schemas

# 04

# Como funciona na prática?

# Inicialização dos serviços

```
# Iniciar o zookeeper
C:\Kafka\application\bin\windows\zookeeper-server-start.bat C:\Kafka\application\config\zookeeper.properties

# Iniciar broker 1
C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server1.properties

# Iniciar broker 2
C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server2.properties

# Iniciar broker 3
C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server3.properties
```

**Window 1 — Prompt de Comando - C:\Kafka\application\bin\windows\zookeeper-server-start.bat C:\Kafka\application\config\zookeeper.properties**

```
[2021-07-04 09:55:53,762] INFO Server environment:user.name=agspf (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,762] INFO Server environment:user.home=C:\Users\agspf (org.apache.zookeeper.server.ZooKeeperServer)
er)
[2021-07-04 09:55:53,763] INFO Server environment:user.dir=C:\ (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,763] INFO Server environment:os.memory.free=493MB (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,763] INFO Server environment:os.memory.max=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,763] INFO Server environment:os.memory.total=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,765] INFO minSessionTimeout set to 6000 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,765] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.server.ZooKeeperServer)
[2021-07-04 09:55:53,765] INFO Created server with tickTime 3000 minSessionTimeout 6000 maxSessionTimeout 60000 datad
ir C:\Kafka\data\zookeeper\version-2 snapdir C:\Kafka\data\zookeeper\version-2 (org.apache.zookeeper.server.ZooKeeper
Server)
[2021-07-04 09:55:53,778] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (o
rg.apache.zookeeper.server.ServerCnxnFactory)
[2021-07-04 09:55:53,781] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 2 selector
 thread(s), 16 worker threads, and 64 kB direct buffers. (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2021-07-04 09:55:53,784] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory
)
[2021-07-04 09:55:53,797] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2021-07-04 09:55:53,800] INFO Snapshotting: 0x0 to C:\Kafka\data\zookeeper\version-2\snapshot.0 (org.apache.zookeepe
r.server.persistence.FileTxnSnapLog)
[2021-07-04 09:55:53,803] INFO Snapshotting: 0x0 to C:\Kafka\data\zookeeper\version-2\snapshot.0 (org.apache.zookeepe
r.server.persistence.FileTxnSnapLog)
[2021-07-04 09:55:53,814] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.serv
er.PrepRequestProcessor)
[2021-07-04 09:55:53,820] INFO Using checkIntervalMs=60000 maxPerMinute=10000 (org.apache.zookeeper.server.ContainerM
anager)
[2021-07-04 09:56:12,421] INFO Creating new log file: log.1 (org.apache.zookeeper.server.persistence.FileTxnLog)
```

**Window 2 — Prompt de Comando - C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server1.properties**

```
[2021-07-04 09:56:13,369] INFO [GroupCoordinator 1]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:13,374] INFO [GroupCoordinator 1]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:13,396] INFO Updated cache from existing <empty> to latest FinalizedFeaturesAndEpoch(features=Featu
res{}, epoch=0). (kafka.server.FinalizedFeatureCache)
[2021-07-04 09:56:13,401] INFO [ProducerId Manager 1]: Acquired new producerId block (brokerId:1,blockStartProducerId
:0,blockEndProducerId:999) by writing to Zk with path version 1 (kafka.coordinator.transaction.ProducerIdManager)
[2021-07-04 09:56:13,402] INFO [TransactionCoordinator id=1] Starting up. (kafka.coordinator.transaction.TransactionC
oordinator)
[2021-07-04 09:56:13,405] INFO [TransactionCoordinator id=1] Startup complete. (kafka.coordinator.transaction.Transac
tionCoordinator)
[2021-07-04 09:56:13,405] INFO [Transaction Marker Channel Manager 1]: Starting (kafka.coordinator.transaction.Transa
ctionMarkerChannelManager)
[2021-07-04 09:56:13,435] INFO [ExpirationReaper-1-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$Expir
edOperationReaper)
[2021-07-04 09:56:13,459] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificatio
nListener$ChangeEventProcessThread)
[2021-07-04 09:56:13,467] INFO [SocketServer listenerType=ZK_BROKER, nodeId=1] Starting socket server acceptors and p
rocessors (kafka.network.SocketServer)
[2021-07-04 09:56:13,472] INFO [SocketServer listenerType=ZK_BROKER, nodeId=1] Started data-plane acceptor and proces
sor(s) for endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServer)
[2021-07-04 09:56:13,472] INFO [SocketServer listenerType=ZK_BROKER, nodeId=1] Started socket server acceptors and pr
ocessors (kafka.network.SocketServer)
[2021-07-04 09:56:13,477] INFO Kafka version: 2.8.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:13,477] INFO Kafka commitId: ebb1d6e21cc92130 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:13,478] INFO Kafka startTimeMs: 1625403373472 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:13,481] INFO [KafkaServer id=1] started (kafka.server.KafkaServer)
[2021-07-04 09:56:13,549] INFO [broker-1-to-controller-send-thread]: Recorded new controller, from now on will use br
oker ANDERSON:9092 (id: 1 rack: null) (kafka.server.BrokerToControllerRequestThread)
```

**Window 3 — Prompt de Comando - C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server2.properties**

```
[2021-07-04 09:56:29,696] INFO [ExpirationReaper-2-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$Expir
edOperationReaper)
[2021-07-04 09:56:29,707] INFO [GroupCoordinator 2]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:29,711] INFO [GroupCoordinator 2]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:29,734] INFO [ProducerId Manager 2]: Acquired new producerId block (brokerId:2,blockStartProducerId
:1000,blockEndProducerId:1999) by writing to Zk with path version 2 (kafka.coordinator.transaction.ProducerIdManager)
[2021-07-04 09:56:29,735] INFO [TransactionCoordinator id=2] Starting up. (kafka.coordinator.transaction.TransactionC
oordinator)
[2021-07-04 09:56:29,738] INFO [TransactionCoordinator id=2] Startup complete. (kafka.coordinator.transaction.Transac
tionCoordinator)
[2021-07-04 09:56:29,739] INFO [Transaction Marker Channel Manager 2]: Starting (kafka.coordinator.transaction.Transa
ctionMarkerChannelManager)
[2021-07-04 09:56:29,762] INFO [ExpirationReaper-2-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$Expir
edOperationReaper)
[2021-07-04 09:56:29,779] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificatio
nListener$ChangeEventProcessThread)
[2021-07-04 09:56:29,787] INFO [SocketServer listenerType=ZK_BROKER, nodeId=2] Starting socket server acceptors and p
rocessors (kafka.network.SocketServer)
[2021-07-04 09:56:29,791] INFO [SocketServer listenerType=ZK_BROKER, nodeId=2] Started data-plane acceptor and proces
sor(s) for endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServer)
[2021-07-04 09:56:29,792] INFO [SocketServer listenerType=ZK_BROKER, nodeId=2] Started socket server acceptors and pr
ocessors (kafka.network.SocketServer)
[2021-07-04 09:56:29,796] INFO Kafka version: 2.8.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:29,797] INFO Kafka commitId: ebb1d6e21cc92130 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:29,797] INFO Kafka startTimeMs: 1625403389792 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:29,799] INFO [KafkaServer id=2] started (kafka.server.KafkaServer)
[2021-07-04 09:56:29,889] INFO [broker-2-to-controller-send-thread]: Recorded new controller, from now on will use br
oker ANDERSON:9092 (id: 1 rack: null) (kafka.server.BrokerToControllerRequestThread)
```

**Window 4 — Prompt de Comando - C:\Kafka\application\bin\windows\kafka-server-start.bat C:\Kafka\application\config\server3.properties**

```
[2021-07-04 09:56:41,938] INFO [ExpirationReaper-3-Rebalance]: Starting (kafka.server.DelayedOperationPurgatory$Expir
edOperationReaper)
[2021-07-04 09:56:41,949] INFO [GroupCoordinator 3]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:41,953] INFO [GroupCoordinator 3]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2021-07-04 09:56:41,978] INFO [ProducerId Manager 3]: Acquired new producerId block (brokerId:3,blockStartProducerId
:2000,blockEndProducerId:2999) by writing to Zk with path version 3 (kafka.coordinator.transaction.ProducerIdManager)
[2021-07-04 09:56:41,979] INFO [TransactionCoordinator id=3] Starting up. (kafka.coordinator.transaction.TransactionC
oordinator)
[2021-07-04 09:56:41,982] INFO [TransactionCoordinator id=3] Startup complete. (kafka.coordinator.transaction.Transac
tionCoordinator)
[2021-07-04 09:56:41,983] INFO [Transaction Marker Channel Manager 3]: Starting (kafka.coordinator.transaction.Transa
ctionMarkerChannelManager)
[2021-07-04 09:56:42,007] INFO [ExpirationReaper-3-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$Expir
edOperationReaper)
[2021-07-04 09:56:42,023] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificatio
nListener$ChangeEventProcessThread)
[2021-07-04 09:56:42,031] INFO [SocketServer listenerType=ZK_BROKER, nodeId=3] Starting socket server acceptors and p
rocessors (kafka.network.SocketServer)
[2021-07-04 09:56:42,036] INFO [SocketServer listenerType=ZK_BROKER, nodeId=3] Started data-plane acceptor and proces
sor(s) for endpoint : ListenerName(PLAINTEXT) (kafka.network.SocketServer)
[2021-07-04 09:56:42,036] INFO [SocketServer listenerType=ZK_BROKER, nodeId=3] Started socket server acceptors and pr
ocessors (kafka.network.SocketServer)
[2021-07-04 09:56:42,041] INFO Kafka version: 2.8.0 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:42,041] INFO Kafka commitId: ebb1d6e21cc92130 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:42,041] INFO Kafka startTimeMs: 1625403402036 (org.apache.kafka.common.utils.AppInfoParser)
[2021-07-04 09:56:42,044] INFO [KafkaServer id=3] started (kafka.server.KafkaServer)
[2021-07-04 09:56:42,126] INFO [broker-3-to-controller-send-thread]: Recorded new controller, from now on will use br
oker ANDERSON:9092 (id: 1 rack: null) (kafka.server.BrokerToControllerRequestThread)
```
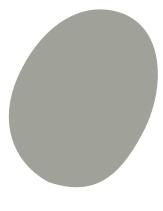
# Criar e descrever tópicos

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --create --topic topic-3-1 --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092
Created topic topic-3-1.

C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-1 --bootstrap-server localhost:9092
Topic: topic-3-1        TopicId: NgZ1I_VRTLy9cyLGZlwJpw PartitionCount: 3        ReplicationFactor: 1    Configs: segment.bytes=1073741824
        Topic: topic-3-1        Partition: 0    Leader: 3       Replicas: 3     Isr: 3
        Topic: topic-3-1        Partition: 1    Leader: 1       Replicas: 1     Isr: 1
        Topic: topic-3-1        Partition: 2    Leader: 2       Replicas: 2     Isr: 2
```

# Criar e descrever tópicos

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --create --topic topic-3-1 --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092
Created topic topic-3-1.

C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-1 --bootstrap-server localhost:9092
Topic: topic-3-1        TopicId: NgZ1I_VRTLy9cyLGZlwJpw PartitionCount: 3        ReplicationFactor: 1    Configs: segment.bytes=1073741824
        Topic: topic-3-1        Partition: 0    Leader: 3       Replicas: 3    Isr: 3
        Topic: topic-3-1        Partition: 1    Leader: 1       Replicas: 1    Isr: 1
        Topic: topic-3-1        Partition: 2    Leader: 2       Replicas: 2    Isr: 2
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --create --topic topic-3-3 --partitions 3 --replication-factor 3 --bootstrap-server localhost:9092
Created topic topic-3-3.

C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3    Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,3,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 3,2,1
```

# E o que acontece se um Broker cair?

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3        Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,3,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 3,2,1
```

# E o que acontece se um Broker cair?

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3      Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,3,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 3,2,1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3      Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1
        Topic: topic-3-3        Partition: 2    Leader: 2       Replicas: 3,2,1 Isr: 2,1
```

# E o que acontece se um Broker cair?

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3        Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,3,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 3,2,1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3        Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1
        Topic: topic-3-3        Partition: 2    Leader: 2       Replicas: 3,2,1 Isr: 2,1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3        Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2,3
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 2       Replicas: 3,2,1 Isr: 2,1,3
```

# E o que acontece se um Broker cair?

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3    Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,3,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 3,2,1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3    Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1
        Topic: topic-3-3        Partition: 2    Leader: 2       Replicas: 3,2,1 Isr: 2,1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3    Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2,3
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 2       Replicas: 3,2,1 Isr: 2,1,3
```

```
C:\>C:\Kafka\application\bin\windows\kafka-topics.bat --describe --topic topic-3-3 --bootstrap-server localhost:9092
Topic: topic-3-3        TopicId: 8gyvzpwzRraVJW9Q-fB_Ug PartitionCount: 3        ReplicationFactor: 3    Configs: segment.bytes=1073741824
        Topic: topic-3-3        Partition: 0    Leader: 1       Replicas: 1,3,2 Isr: 1,2,3
        Topic: topic-3-3        Partition: 1    Leader: 2       Replicas: 2,1,3 Isr: 2,1,3
        Topic: topic-3-3        Partition: 2    Leader: 3       Replicas: 3,2,1 Isr: 2,1,3
```
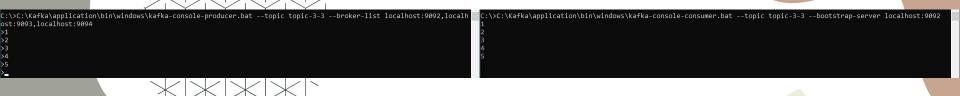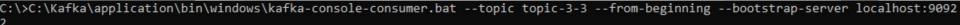
# Produtores e consumidores

```
C:\>C:\Kafka\application\bin\windows\kafka-console-producer.bat --topic topic-3-3 --broker-list localhost:9092,localh
ost:9093,localhost:9094
>1
>2
>3
>4
>5
>
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --bootstrap-server localhost:9092
1
2
3
4
5
```

# Produtores e consumidores

```
C:\>C:\Kafka\application\bin\windows\kafka-console-producer.bat --topic topic-3-3 --broker-list localhost:9092,localh
ost:9093,localhost:9094
>1
>2
>3
>4
>5
>_
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --bootstrap-server localhost:9092
1
2
3
4
5
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --bootstrap-server localhost:9092
_
```

# Produtores e consumidores

```
C:\>C:\Kafka\application\bin\windows\kafka-console-producer.bat --topic topic-3-3 --broker-list localhost:9092,localh
ost:9093,localhost:9094
>1
>2
>3
>4
>5
>
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --bootstrap-server localhost:9092
1
2
3
4
5
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --bootstrap-server localhost:9092
_
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --from-beginning --bootstrap-server localhost:9092
2
3
1
4
5
```

# Produtores e consumidores

```
C:\>C:\Kafka\application\bin\windows\kafka-console-producer.bat --topic topic-3-3 --broker-list localhost:9092,localh
ost:9093,localhost:9094
>1
>2
>3
>4
>5
>6
>7
>8
>9
>10
>
```

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --group group-1 --bootstrap-server
localhost:9092
2
3
8
9
```

Prompt de Comando - C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --group group-1 --bootstrap-server local...

```
C:\>C:\Kafka\application\bin\windows\kafka-console-consumer.bat --topic topic-3-3 --group group-1 --bootstrap-server
localhost:9092
1
4
5
6
7
10
```

# Produtor e consumidores no mesmo grupo

```
C:\>C:\Kafka\application\bin\windows\kafka-consumer-groups.bat --group group-1 --describe --bootstrap-server localhost:9092

GROUP       TOPIC        PARTITION   CURRENT-OFFSET   LOG-END-OFFSET   LAG   CONSUMER-ID                                                  HOST            CLIENT-ID
group-1     topic-3-3    0           9                9                0     consumer-group-1-1-6ec31caf-d44e-4325-9452-26c07328492c      /192.168.0.14   consumer-group-1-1
group-1     topic-3-3    1           5                5                0     consumer-group-1-1-6ec31caf-d44e-4325-9452-26c07328492c      /192.168.0.14   consumer-group-1-1
group-1     topic-3-3    2           7                7                0     consumer-group-1-1-d3f3d6ae-b82c-4248-9238-18b222a9688a      /192.168.0.14   consumer-group-1-1
```

# Produtor e consumidores no mesmo grupo

```
C:\>C:\Kafka\application\bin\windows\kafka-consumer-groups.bat --group group-1 --describe --bootstrap-server localhost:9092

GROUP      TOPIC       PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG      CONSUMER-ID                                        HOST            CLIENT-ID
group-1    topic-3-3   0          9               9               0        consumer-group-1-1-6ec31caf-d44e-4325-9452-26c07328492c  /192.168.0.14   consumer-group-1-1
group-1    topic-3-3   1          5               5               0        consumer-group-1-1-6ec31caf-d44e-4325-9452-26c07328492c  /192.168.0.14   consumer-group-1-1
group-1    topic-3-3   2          7               7               0        consumer-group-1-1-d3f3d6ae-b82c-4248-9238-18b222a9688a  /192.168.0.14   consumer-group-1-1
```

```
C:\>C:\Kafka\application\bin\windows\kafka-consumer-groups.bat --group group-1 --describe --bootstrap-server localhost:9092

GROUP      TOPIC       PARTITION  CURRENT-OFFSET  LOG-END-OFFSET  LAG      CONSUMER-ID                                        HOST            CLIENT-ID
group-1    topic-3-3   1          5               5               0        consumer-group-1-1-c3153fd8-ea31-49f1-a546-a2fad4f318b6  /192.168.0.14   consumer-group-1-1
group-1    topic-3-3   2          7               7               0        consumer-group-1-1-d3f3d6ae-b82c-4248-9238-18b222a9688a  /192.168.0.14   consumer-group-1-1
group-1    topic-3-3   0          9               9               0        consumer-group-1-1-6ec31caf-d44e-4325-9452-26c07328492c  /192.168.0.14   consumer-group-1-1
```

# Produtor em Java

```java
public class ProducerCreator {
    public static String KAFKA_BROKERS = "localhost:9092";
    public static String TOPIC_NAME = "topic-3-3";
    public static int MESSAGE_COUNT = 1000;

    public static Producer<Long, String> createProducer() {
        Properties props = new Properties();
        props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, KAFKA_BROKERS);
        props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, LongSerializer.class.getName());
        props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, StringSerializer.class.getName());
        return new KafkaProducer<>(props);
    }
}

static void runProducer() {
    Producer<Long, String> producer = ProducerCreator.createProducer();

    for (int index = 0; index < IKafkaConstants.MESSAGE_COUNT; index++) {
        ProducerRecord<Long, String> record = new ProducerRecord<Long, String>(ProducerCreator.TOPIC_NAME, "This is record " + index);
        try {
            RecordMetadata metadata = producer.send(record).get();
            System.out.println("Record sent with key " + index + " to partition " + metadata.partition() + " with offset " + metadata.offset());
        } catch (ExecutionException e) {
            System.out.println("Error in sending record");
            System.out.println(e);
        } catch (InterruptedException e) {
            System.out.println("Error in sending record");
            System.out.println(e);
        }
    }
}
```

# Consumidor em Java

```java
public class ConsumerCreator {
    public static String KAFKA_BROKERS = "localhost:9092";
    public static String GROUP_ID = "group-1";
    public static String TOPIC_NAME = "topic-3-3";

    public static Consumer<Long, String> createConsumer() {
        Properties props = new Properties();
        props.put(ConsumerConfig.BOOTSTRAP_SERVERS_CONFIG, KAFKA_BROKERS);
        props.put(ConsumerConfig.GROUP_ID_CONFIG, GROUP_ID);
        props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG, LongDeserializer.class.getName());
        props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG, StringDeserializer.class.getName());
        Consumer<Long, String> consumer = new KafkaConsumer<>(props);
        consumer.subscribe(Collections.singletonList(TOPIC_NAME));
        return consumer;
    }
}

static void runConsumer() {
    Consumer<Long, String> consumer = ConsumerCreator.createConsumer();
    int noMessageFound = 0;
    while (true) {
        ConsumerRecords<Long, String> consumerRecords = consumer.poll(1000);
        if (consumerRecords.count() == 0) {
            noMessageFound++;
            if (noMessageFound > IKafkaConstants.MAX_NO_MESSAGE_FOUND_COUNT)
                break;
        }
        consumerRecords.forEach(record -> {
            System.out.println("Record Key " + record.key());
            System.out.println("Record value " + record.value());
            System.out.println("Record partition " + record.partition());
            System.out.println("Record offset " + record.offset());
        });
        consumer.commitAsync();
    }
    consumer.close();
}
```

# Obrigado!

# Referências

- https://docs.confluent.io/platform/current/schema-registry/index.html

- https://kafka.apache.org/documentation/

- https://kafka.apache.org/quickstart