

Testes em tópicos de Kafka:
uma sugestão de como
realizar

Aluna: Isadora Rocha de Paula



Agenda

- Motivação
- Apis
- Arquitetura de microsserviços
- Arquitetura orientada a eventos
- Stream
- Kafka
- Testes em tópicos de Kafka



Motivação

- São poucos os QAs que sabem como testar back-end
- Não conhecem Kafka ou arquitetura orientada a eventos
- Não sabem testar Kafka



APIs

É a capacidade de dois sistemas interagirem sem saber necessariamente como o outro funciona, seja para receber ou enviar dados



APIs públicas

- Google Maps
- Google captcha



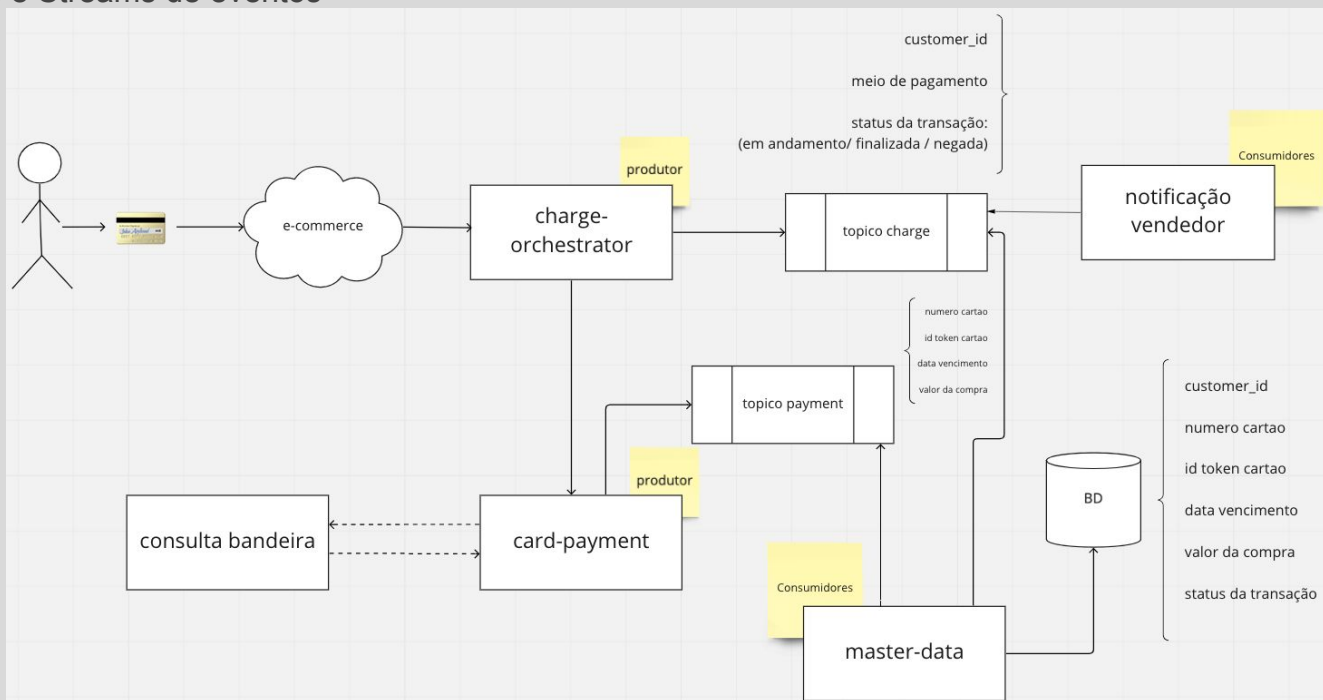
Arquitetura de Microsserviços

- Estilo de arquitetura de software
- É a separação de serviços por especialidade ou responsabilidade
 - Ex.: Site de e-commerce
- Vantagens
 - Paralelização do desenvolvimento da aplicação entre diferentes equipes de desenvolvimento
 - Tempo de implementação reduzido
 - Facilidade de implementação
 - Independência entre serviços



Arquitetura orientada a eventos

- Estilo de arquitetura voltada para a comunicação assíncrona dos componentes
- Eventos e Streams de eventos
- Ex.:



Streaming

- Stream = fluxo / Streaming = transmissão
- Transmissão contínua de dados
 - Spotify
 - Netflix
- Ex.:
 - Avenida que sempre passa carros = fluxo
 - Carro = Dado da mensagem
 - Passagem do carro = geração do evento
- Stream processing: análise dos dados
 - Métricas





Kafka

- Como surgiu
- Definições e funcionalidades
- Como funciona



Como surgiu

- Surgiu em 2010
- Coletas de métricas do LinkedIn



Como surgiu

- Publicação mensagens em lote de hora em hora em formato XML
- Processamento de dados de maneira off line
- Análise cara



Como surgiu

- Sistema capaz de:
 - ter um processamento de alta performance
 - capaz de lidar com grande quantidade de dados
 - processar dados em tempo real
 - monitorar e rastrear eventos
 - separar produtores de consumidores



Definições e funcionalidades

Kafka é um sistema de mensagem que permite a escrita, leitura, armazenagem e processamento de dados de streaming, mantendo o controle de eventos.



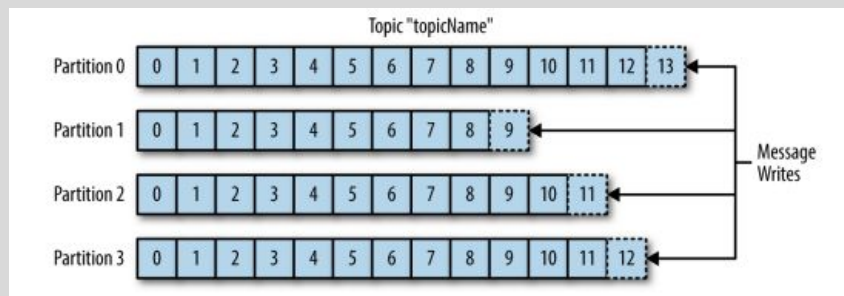
Definições e funcionalidades

- Pode ser usado para:
 - rastreamento de atividades
 - enviar notificações aos usuários
 - colher métricas para monitoramento
 - emitir alertas
 - auxiliar na atualização de banco de dados



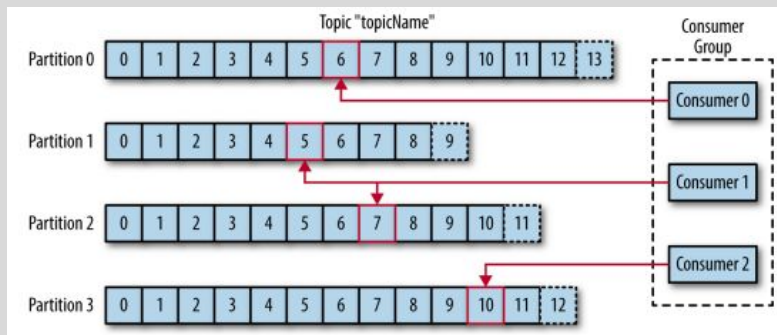
Como funciona

- Armazena e processa fluxo de dados continuamente
- Processa dados contínuos e de baixa latência
- Estrutura:
 - produtores
 - tópicos
 - partições
 - consumidores
 - offset



Como funciona

- Armazena e processa fluxo de dados continuamente
- Processa dados contínuos e de baixa latência
- Estrutura:
 - produtores
 - tópicos
 - partições
 - consumidores
 - offset



Como funciona

- Armazena e processa fluxo de dados continuamente
- Processa dados contínuos e de baixa latência
- Estrutura:
 - brokers
 - servidor do kafka
 - guardar as mensagens criadas nos tópicos em disco
 - marcação do offset
 - atende os consumidores na busca das mensagens por partições
- [Simulação](#)



Testando tópicos do Kafka

- Como identificar qual ação gera o evento?
- Como saber como identificar o evento?
- Para testes manuais: Kafka Magic
- Para automação de testes: ruby-kafka



Kafka Magic

O Kafka Magic é uma ferramenta de interface gráfica, desenvolvida pela própria Apache especificamente para trabalhar com os clusters do Kafka. É capaz de localizar e exibir mensagens, transformar e mover mensagens entre tópicos, revisar e atualizar esquemas, gerenciar tópicos e automatizar tarefas complexas.



Ruby

- Criada em 1995 por Yukihiro Matsumot
- É uma linguagem orientada a objetos
- As bibliotecas são chamadas de “gem”
- As principais gems utilizadas neste trabalho foram
 - cucumber
 - ruby-kafka
 - report_builder



Cucumber

- O cucumber é uma ferramenta que permite o automatizador de testes usar o Behaviour-Driven Development (BDD) em seus testes, o que traduzido significa Desenvolvimento Orientado ao Comportamento
- É escrito em Gherking
 - Dado
 - Quando
 - Então

```
2
3  @teste
4  >  Funcionalidade: Consulta de mensagens em tópico do Kafka
5
6  >  Esquema do Cenário: Consultar a publicação em mensagens no tópico do Kafka local com sucesso
7      Dado que realize o POST de uma mensagem qualquer
8      Então valido que houve o consumo dessa mensagem no tópico do kafka local em <seg>
9
10     Exemplos:
11         | seg |
12         | 20.0 |
13         | 0.0001 |
14
```



Ruby-Kafka

```
1 class KafkaTest
2   # require "kafka"
3   def read_message
4     #caminho do broker do kafka
5     kafka = Kafka.new(["localhost:9092"])
6
7     # ID do grupo de consumidores
8     consumer = kafka.consumer(group_id: "my-consumer")
9
10    #nome do topico a ser consumido
11    consumer.subscribe("tcc_topic")
12
13    #variavel responsavel por controle de tempo que o consumidor ficara em funcionamento
14    time_start = Time.new
15
16    #loop para leitura das mensagens do kakfa
17    consumer.each_message do |message|
18
19      #acessa o valor da mensagem do tópico
20      message = message.value
21
22      #printa no console a mensagem que foi consumida
23      puts message

```

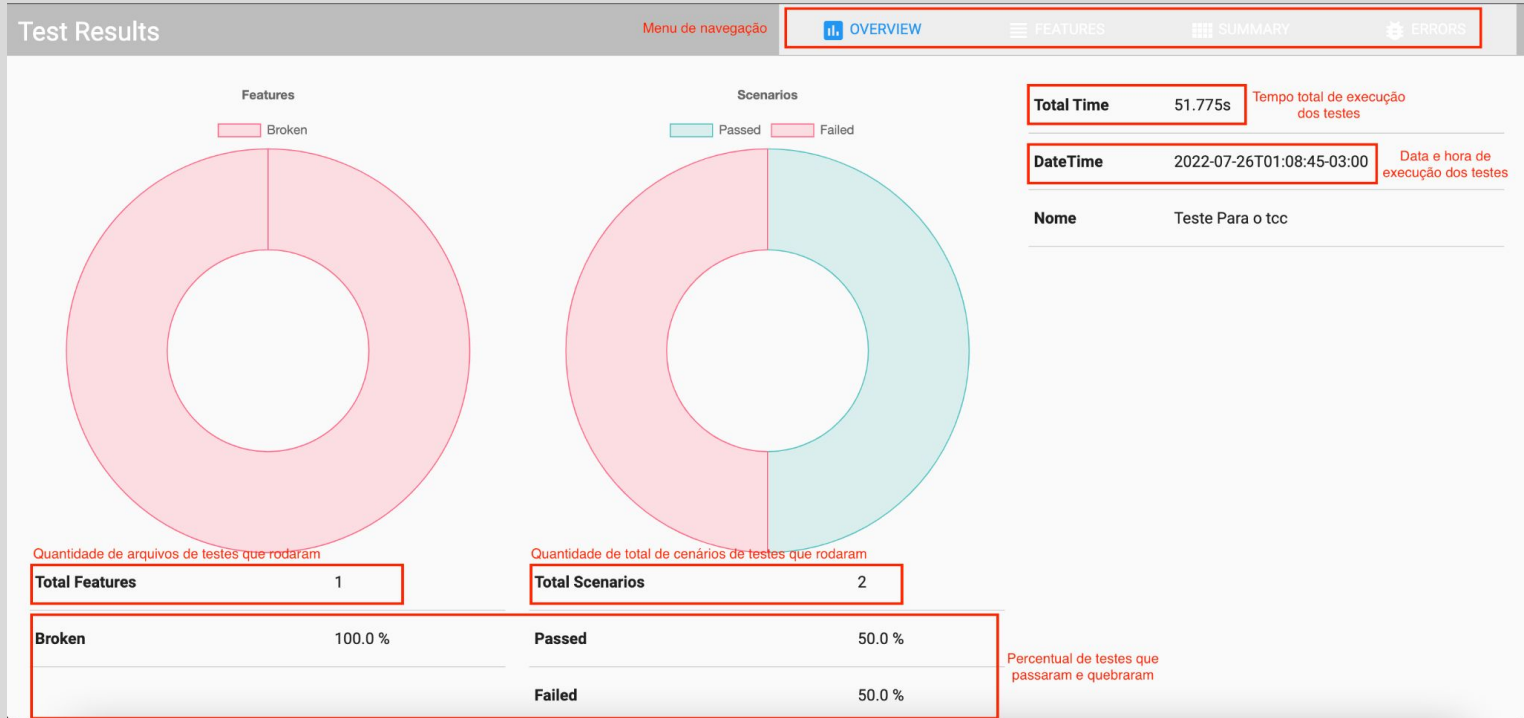


Ruby-Kafka

```
23      #printa no console a mensagem que foi consumida
24      puts message
25      #entra na condicao se a mensagem lida for igual a que for enviada na requisicao
26      if message.eql?(message)
27
28          # o loop eh interrompido e retorna o valor true e a mensagem encontrada
29          return true, message
30
31      #entra na condicao se der o tempo limite de 3 segundos de funcionamento do considor
32      elsif Time.new - time_start > temp
33
34          #comando de parada do consumidor
35          consumer.stop
36
37          #retorna o valor false e a ultima mensagem que foi consumida
38          return false, message
39      end
40  end
41 end
42 end
```



Report_builder





Perguntas?





Obrigada

