



# Apache Kafka

Mateus Lucas Cruz Brandt

2024/1 CEAVI Engenharia de Software Orientada a Serviços

# Contextualização

# Mensageria

Abordagem fundamental no desenvolvimento de sistemas distribuídos, permitindo comunicação eficiente entre diferentes componentes ou serviços

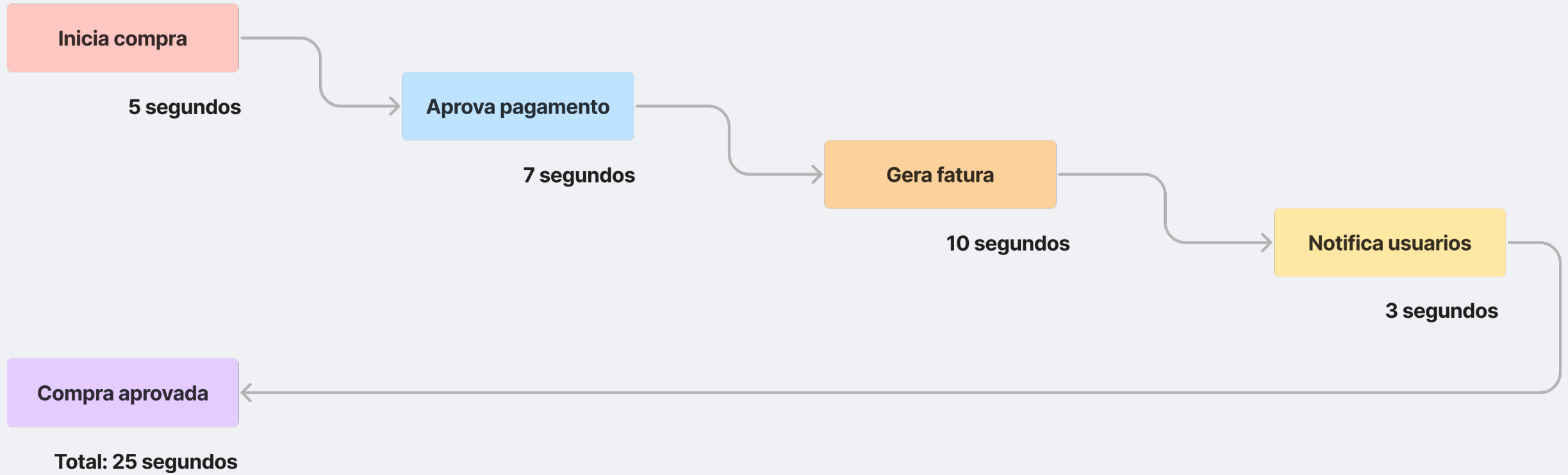
Em um ambiente distribuído, onde diversos elementos do sistema podem estar em máquinas diferentes, a mensageria desempenha um papel crucial na troca de informações assíncronas

# Problema

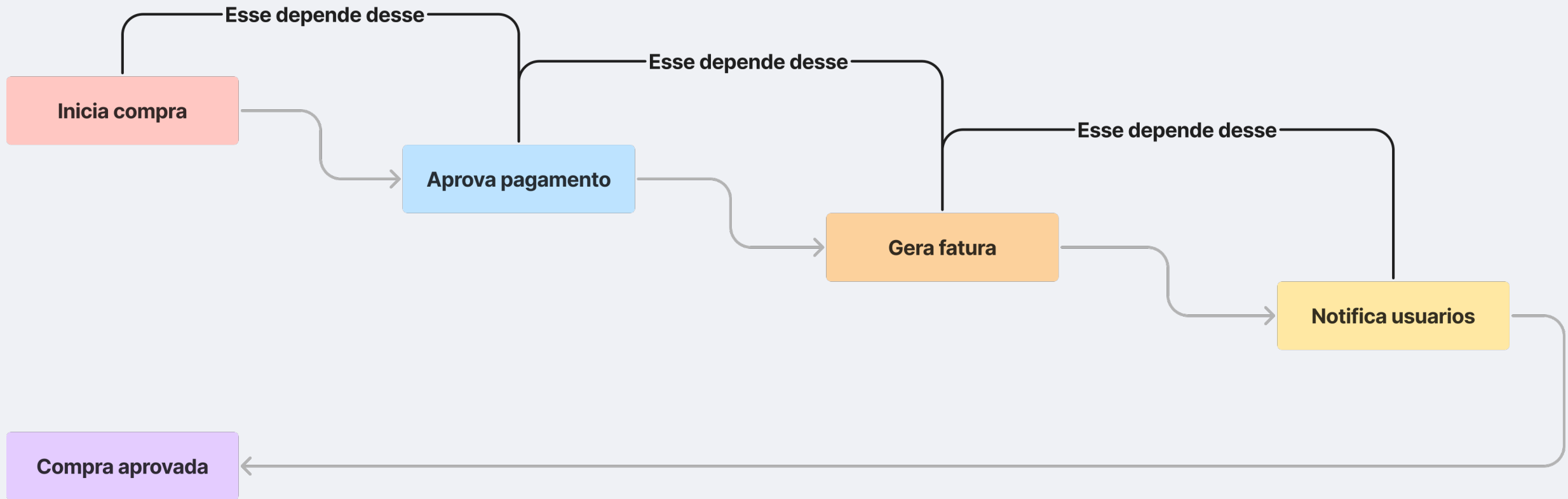
# Desacoplamento em Sistemas

À medida que os sistemas escalam em tamanho e complexidade, surge a necessidade crítica de desacoplamento. Este desafio se manifesta em duas dimensões principais: temporal e arquitetônica.

# Temporal



# Arquitetural



# Como mensageria resolve?

## Comunicação Assíncrona

Permite que os sistemas se comuniquem de forma assíncrona, ou seja, os componentes não precisam estar ativos simultaneamente para trocar informações

## Desacoplamento

Sistemas podem ser desacoplados, o que significa que os produtores e consumidores de mensagens não precisam estar cientes uns dos outros. Isso proporciona flexibilidade e escalabilidade (“GoF/Observer”)

## Confiança e Resiliência

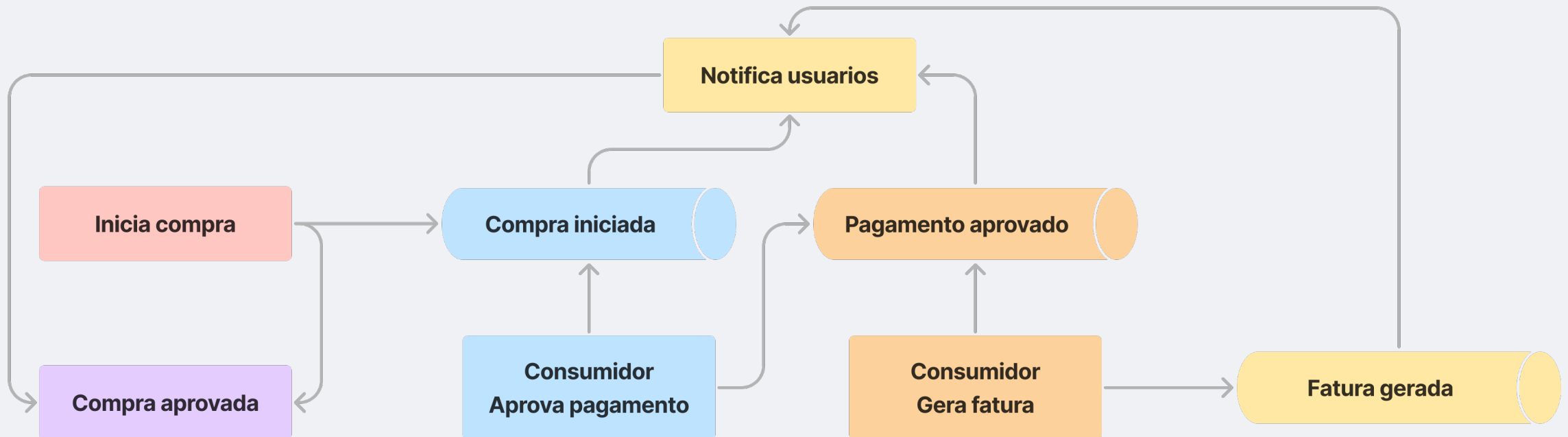
Ajuda a construir sistemas mais resilientes, uma vez que as mensagens podem ser armazenadas temporariamente e reprocessadas em caso de falhas

## Escalabilidade

Permite a escalabilidade horizontal, uma vez que novos componentes podem ser adicionados sem afetar diretamente os existentes, facilitando a adaptação a cargas de trabalho variáveis



# Como mensageria resolve?

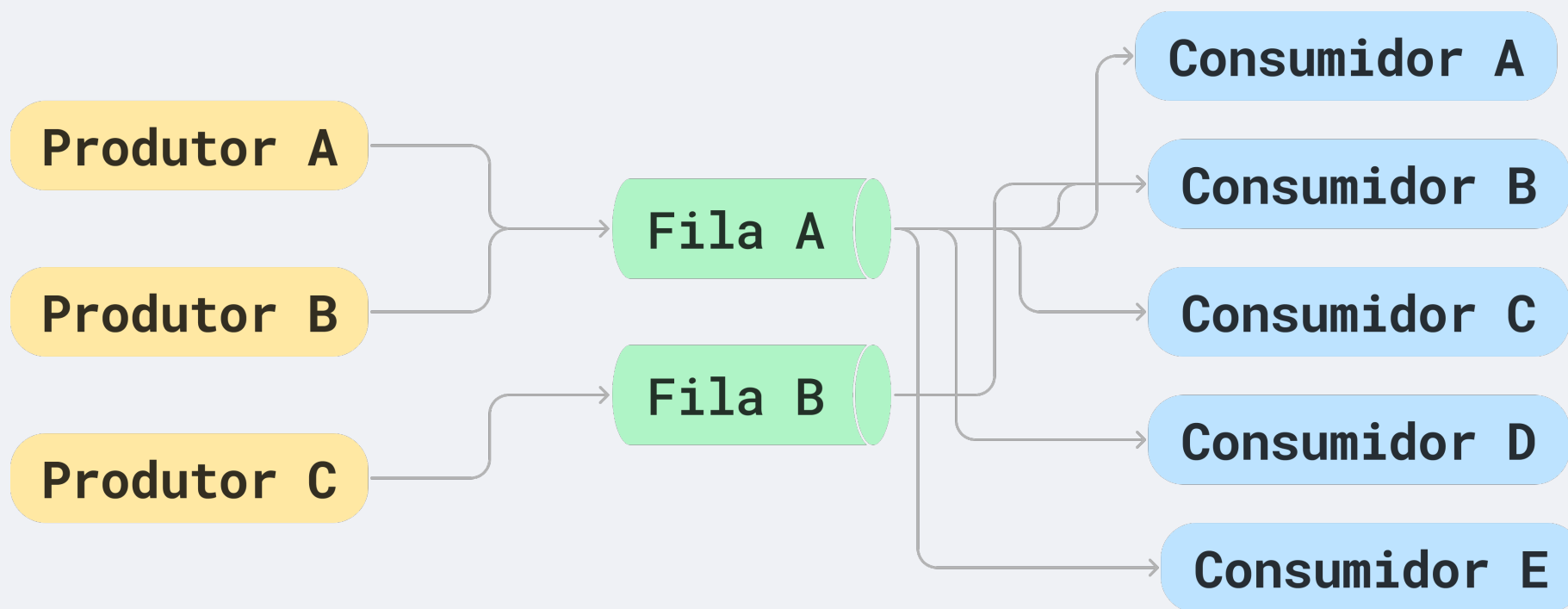




Apache no resgate

# Visão Geral

O Apache Kafka é uma plataforma de streaming distribuída, projetada para lidar com grandes volumes de dados em tempo real.



# Como funciona

## **Tópicos**

Os dados são organizados em tópicos (filas). Onde cada mensagem é um par chave-valor.

## **Produtores**

São aplicações responsáveis por alimentar os tópicos. Podem escrever para um ou mais tópicos.

## **Consumidores**

São aplicações que "escutam" tópicos. Podem processar mensagem a mensagem ou então em lotes. Também são responsáveis por gerenciar quais as mensagens que já foram consumidas

# Como funciona

## Brokers

São os "servidores" responsáveis por gerenciar os dados do Kafka

## Partições

Cada tópico está dividido em partições. Elas permitem escala horizontal.

## Réplicas

Para auxiliar na segurança de dados, cada partição pode ter réplicas. A réplica principal, escreve nas demais. Se uma das réplicas cair, uma nova principal é eleita

# Como funciona

## **ZooKeeper**

Responsável por gerenciar e coordenar os brokers.

## **Grupos de consumidor**

Consumidores podem ser gerenciados em grupos para organizar o consumo de mensagem. Permite escala horizontal de consumidores

## **Escalabilidade**

Kafka permite escala tanto horizontal quanto vertical. Novos brokers podem ser adicionados para aumentar capacidade, e partições para distribuir carga entre brokers.

# Vantagens

**Escalabilidade**

**Alta taxa de transferência e baixa latência**

**Persistência**

**Tolerância a falhas**

**Flexibilidade na modelagem de dados**

**Processamento de “streams”**

**Integração no ecossistema Apache**

**Gerenciamento de versões dos tópicos**

# Desvantagens

**Complexidade de configuração**

**Overhead de gerenciamento**

**Armazenamento e recursos necessários**

**Latência de entrega não garantida**

**Complexidade de desenvolvimento**

**Curva de aprendizado**

**Tamanho mínimo do cluster**



# Caso de Uso

Consumindo mais de 1 bilhão de mensagens por dia no Ifood



# Empresas que usam

Powered By

adidas



NETFLIX

