

# Levantamento de Requisitos para Rede Kafka com .NET Core 8

## Descrição do Sistema

**Rede Kafka** é uma plataforma de mensageria distribuída de alto desempenho, construída sobre o Apache Kafka, projetada para permitir a troca eficiente de mensagens entre sistemas e microserviços em tempo real. Com a utilização do **.NET Core 8**, a **Rede Kafka** oferece uma solução escalável, resiliente e segura para a comunicação assíncrona e processamento de grandes volumes de dados. Ideal para ambientes de alta performance e escalabilidade, como e-commerce, fintechs, IoT, big data e aplicações em tempo real, a **Rede Kafka** garante alta disponibilidade e entrega confiável de mensagens.

A **Rede Kafka** é voltada para cenários que exigem integração de sistemas distribuídos, oferecendo funcionalidades como persistência de mensagens, tolerância a falhas, enfileiramento, processamento paralelo e controle de fluxo. Através da arquitetura baseada em tópicos e partições do Kafka, o sistema proporciona resiliência e permite a escalabilidade tanto na publicação quanto no consumo de mensagens.

## Requisitos Funcionais

### 1. Envio e Consumo de Mensagens

#### Envio Assíncrono de Mensagens para Tópicos

O sistema permitirá que as aplicações enviem mensagens de maneira assíncrona para tópicos Kafka, sem bloquear a execução da aplicação.

#### Recepção Imediata de Mensagens

O sistema permitirá que as mensagens sejam consumidas em tempo real, com baixa latência, de tópicos Kafka pelos consumidores.

#### Garantia de Entrega de Mensagens (At-Least-Once)

O sistema garantirá a entrega de mensagens utilizando o modelo **at-least-once**, para garantir que todas as mensagens sejam eventualmente entregues, mesmo em caso de falhas temporárias.

### **Processamento em Lote de Mensagens**

O sistema permitirá o processamento de múltiplas mensagens em lote, otimizando o uso de recursos e aumentando a eficiência do processamento.

### **Mensagens de Prioridade Configuráveis**

O sistema permitirá que as mensagens sejam enviadas com diferentes níveis de prioridade, garantindo que mensagens de alta prioridade sejam processadas primeiro.

### **Criptografia de Mensagens em Trânsito e Repouso**

Todas as mensagens enviadas e recebidas serão criptografadas tanto em trânsito (via TLS) quanto em repouso (utilizando AES-256) para garantir a confidencialidade dos dados.

### **Persistência de Mensagens Não Processadas**

Mensagens que não forem processadas corretamente serão mantidas no Kafka até que possam ser consumidas e processadas com sucesso.

### **Redirecionamento de Mensagens Falhas para DLQ (Dead Letter Queue)**

O sistema moverá mensagens falhas para uma fila de Dead Letter Queue (DLQ), onde poderão ser analisadas e reprocessadas conforme necessário.

### **Definição de Timeout para Processamento de Mensagens**

O sistema permitirá a configuração de tempos limite para o processamento de mensagens, prevenindo que mensagens fiquem presas em processos falhos ou demorados.

### **Roteamento Dinâmico de Mensagens**

O sistema permitirá o roteamento dinâmico de mensagens para diferentes tópicos ou consumidores com base no conteúdo das mensagens ou nas configurações do sistema.

## 2. Monitoramento e Auditoria

### **Monitoramento de Tópicos e Partições em Tempo Real**

O sistema permitirá monitorar em tempo real o status dos tópicos e partições, incluindo a quantidade de mensagens enfileiradas, tempo de espera e taxa de erro.

### **Logs Detalhados de Processamento de Mensagens**

O sistema gerará logs detalhados de todas as operações de envio e consumo de mensagens, para auditoria e análise de falhas.

### **Alertas para Falhas no Processamento de Mensagens**

O sistema enviará alertas automáticos para administradores sempre que ocorrerem falhas no processamento de mensagens, como falhas de entrega ou erro de timeout.

### **Relatórios de Performance de Tópicos e Consumidores**

O sistema deverá gerar relatórios de performance dos tópicos e dos consumidores, com métricas como latência média, throughput, tempo de processamento e taxa de falhas.

### **Auditoria Completa do Ciclo de Vida das Mensagens**

O sistema deve manter um histórico completo do ciclo de vida das mensagens, desde a publicação até o consumo final, incluindo status, timestamps e detalhes de falhas, para rastreabilidade e conformidade regulatória.

### **Monitoramento de Acknowledgments (ACK/NACK)**

O sistema permitirá monitorar os ACKs (confirmações) e NACKs (não confirmações) de mensagens em tempo real, permitindo o rastreamento do status de entrega das mensagens.

### **Relatórios Detalhados da Dead Letter Queue (DLQ)**

O sistema deve fornecer relatórios detalhados sobre as mensagens na DLQ, incluindo as razões pelas quais as mensagens foram movidas para lá e ações necessárias para reproprocessamento.

### **Notificação de Falhas no Reprocessamento**

O sistema enviará notificações caso uma mensagem falha ao ser reproprocessada após ser movida para a DLQ.

## 3. Configuração e Integração

### **Configuração Dinâmica de Tópicos e Partições**

O sistema permitirá a configuração dinâmica de tópicos e partições do Kafka, sem necessidade de reiniciar o serviço ou interromper as operações.

### **Integração com Ferramentas de Monitoramento como Prometheus e Grafana**

O sistema se integrará com ferramentas como **Prometheus** e **Grafana** para coletar e exibir métricas de desempenho de forma detalhada e em tempo real.

### **Suporte à Criação e Gerenciamento de Tópicos Kafka**

O sistema permitirá a criação, modificação e remoção de tópicos diretamente pela interface administrativa, facilitando o gerenciamento.

### **Suporte a Vários Brokers Kafka**

O sistema suportará múltiplos brokers Kafka para garantir alta disponibilidade, balanceamento de carga e tolerância a falhas.

### **Integração com Ferramentas de Gerenciamento do Kafka**

O sistema se integrará com ferramentas de gerenciamento Kafka como **Kafka Manager** ou **Confluent Control Center** para facilitar a administração dos brokers e tópicos.

## **4. Escalabilidade e Resiliência**

### **Escalabilidade Horizontal de Consumidores**

O sistema permitirá escalar horizontalmente os consumidores, garantindo que a carga de mensagens seja distribuída adequadamente quando o volume de tráfego crescer.

### **Failover Automático de Brokers Kafka**

O sistema implementará failover automático para brokers Kafka, garantindo a continuidade das operações mesmo em caso de falha de um broker.

### **Replicação de Dados em Múltiplos Brokers**

O sistema garantirá a replicação de dados entre múltiplos brokers Kafka, aumentando a disponibilidade e a resiliência do sistema.

### **Tolerância a Falhas nas Partições de Tópicos**

Em caso de falhas nas partições de tópicos, o sistema deverá redirecionar automaticamente as mensagens para outras partições ou brokers para garantir a continuidade do processamento.

### **Escalabilidade Independente para Publicadores e Consumidores**

O sistema permitirá a escalabilidade independente de publicadores e consumidores, de acordo com as necessidades de carga e volume de dados de cada componente.

### **Suporte a Processamento Paralelo de Mensagens**

O sistema suportará o processamento paralelo de mensagens, permitindo que múltiplos consumidores ou instâncias do sistema consumam mensagens simultaneamente para melhorar a performance.

### **Capacidade de Processamento de Grandes Volumes de Mensagens**

O sistema será projetado para processar grandes volumes de mensagens com alta eficiência, utilizando técnicas de particionamento e balanceamento de carga.

## **Requisitos Não Funcionais**

### **1. Desempenho**

#### **Baixa Latência nas Operações de Envio e Consumo**

O sistema deverá garantir uma latência inferior a 10 ms nas operações de envio e consumo de mensagens.

#### **Alto Throughput**

O sistema será capaz de processar milhões de mensagens por segundo, mesmo sob alta carga.

#### **Escalabilidade Sob Demanda**

O sistema deverá permitir que novos consumidores e publicadores sejam adicionados de forma transparente, sem impacto significativo na performance.

#### **Alta Disponibilidade**

O sistema deverá garantir alta disponibilidade (99,99%) em todas as operações de envio, recebimento e processamento de mensagens.

#### **Resiliência a Falhas**

O sistema deverá ser resiliente a falhas de brokers ou consumidores, com failover automático e backup de dados persistentes.

### **2. Segurança**

#### **Criptografia de Mensagens**

Todas as mensagens enviadas e recebidas pelo sistema deverão ser criptografadas em trânsito e em repouso para garantir a proteção de dados sensíveis.

#### **Autenticação e Autorização**

O sistema deverá implementar autenticação robusta e autorização granular

para garantir que apenas usuários ou sistemas autorizados possam enviar ou consumir mensagens.

### **Controle de Acesso Baseado em Roles**

O sistema deverá permitir o controle de acesso baseado em roles (RBAC), garantindo que diferentes usuários ou aplicações tenham acesso apenas às funcionalidades e dados necessários.