Roteiro de Atividade Prática

Nome: Turma: .

**Atividade: Gerenciamento eficiente de filas e tópicos**

**Tempo estimado:** 30 minutos.

**Lista de materiais**

* Computador com internet;
* Caderno para anotações;
* 1 caneta.

**Objetivo:**

O objetivo desta atividade é desenvolver um algoritmo que gerencie a retenção e reprocessamento de mensagens falhas até um limite de tentativas, após o qual as mensagens são movidas para uma DLQ.

**Contexto:**

Em sistemas de mensageria, é comum que algumas mensagens falhem na entrega devido a problemas técnicos, como falhas de rede ou erros no processamento. Para garantir a robustez e confiabilidade do sistema, é essencial gerenciar essas falhas de forma eficiente. O uso de Dead Letter Queues (DLQs) permite que mensagens que não podem ser entregues após várias tentativas sejam separadas do fluxo normal para análise e tratamento posterior.

**Enunciado:**

Elabore um algoritmo que implemente o processo de tratamento e reprocessamento de mensagens em um sistema de mensageria, utilizando Dead Letter Queues para gerenciar mensagens que falham repetidamente na entrega. O algoritmo deve descrever como as mensagens são tratadas até um limite de tentativas antes de serem encaminhadas para a DLQ. Acompanhe as etapas detalhadas abaixo para desenvolver o algoritmo.

1. **Etapas para desenvolver o algoritmo**

**1.1 Definição de estruturas:**

* Uma fila principal onde todas as mensagens são inicialmente postadas;
* Uma Dead Letter Queue para armazenar mensagens que excederam o limite de tentativas de entrega.

**1.2 Processo de tratamento de mensagens:**

* Entradas: Mensagem recebida, número máximo de tentativas permitidas.

**1.3 Processo:**

* Tente entregar a mensagem;
* Se a entrega falhar, incremente um contador de tentativas na mensagem;
* Verifique se o contador de tentativas excedeu o limite;
* Se não excedeu, reenfileire a mensagem na fila principal para uma nova tentativa;
* Se excedeu, mova a mensagem para a DLQ;
* Saída: Mensagem entregue com sucesso, mensagem reenfileirada para nova tentativa ou mensagem movida para a DLQ.

**1.4 Manutenção da DLQ:**

* Regularmente, analise as mensagens na DLQ para identificar problemas comuns e potenciais melhorias no sistema de mensageria ou nos produtores de mensagens;
* Determine se as mensagens na DLQ podem ser corrigidas manualmente ou por meio de ajustes automáticos e, então, possivelmente, reenviadas para a fila principal.

1. **Passo a passo da implementação:**

**2.1 Configuração das filas:**

* Estabeleça a fila principal e a DLQ dentro do sistema de mensageria;
* Configure as propriedades da fila, como tamanho, tempo de vida das mensagens, e políticas de retenção.

**2.2 Desenvolvimento do módulo de processamento de mensagens:**

* Implemente uma função que processe mensagens da fila principal;
* Adicione lógica para tentativas de entrega e tratamento de falhas.

**2.3 Implementação do contador de tentativas:**

* Integre um contador de tentativas em cada mensagem;
* Assegure que o contador seja incrementado apropriadamente a cada falha de entrega.

**2.4 Integração com a DLQ:**

* Implemente a lógica de movimentação de mensagens para a DLQ após o excedente de tentativas;
* Garanta que a DLQ seja monitorada e gerenciada conforme necessário.

**2.5 Testes e validação:**

* Simule falhas de entrega para testar a robustez do algoritmo;
* Verifique se as mensagens são corretamente movidas para a DLQ após o limite de tentativas;
* Teste a recuperação e o reprocessamento de mensagens da DLQ.

**2.6 Monitoramento e análise:**

* Implemente ferramentas de monitoramento para acompanhar o número de mensagens falhas e a utilização da DLQ;
* Analise periodicamente as causas das falhas para melhorar continuamente o sistema de mensageria.

Este algoritmo oferece uma abordagem estruturada para gerenciar falhas em sistemas de mensageria, aumentando a confiabilidade e eficiência do sistema ao separar e tratar mensagens problemáticas de forma sistemática.

**Procedimento experimental**

1. A partir do contexto e das etapas desenvolvidas, descreva o algoritmo obtido na atividade e registre no AVA:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |