

17/03/2023

# UM SISTEMA DE MONITORAMENTO ORIENTADO A EVENTOS – PARTE 1

## 1. REGRAS GERAIS

- i. O projeto pode ser desenvolvido em dupla. A data de entrega preliminar da primeira parte é 08/05/2023;
- ii. As duplas farão uma apresentação simples do projeto para os colegas, durante a aula (10 a 15 minutos de apresentação).

## 2. DESCRIÇÃO

Um sistema de monitoramento orientado a eventos tem como objetivo o acompanhamento contínuo de recursos por meio de eventos. De maneira simplificada, o funcionamento geral de um sistema deste tipo segue o seguinte padrão: recursos produzem eventos de interesse que devem ser continuamente consumidos e monitorados em tempo real; quando situações de interesse são detectadas, ações devem ser tomadas (por exemplo, notificações, execução de serviços, produção de novos eventos, etc.).

Vocês devem implementar um sistema desse tipo, usando a plataforma Apache Kafka como infraestrutura de base. Portanto, vocês precisam estabelecer uma arquitetura que seja consistente com a arquitetura Kafka, i.e., com produtores e consumidores de eventos, e Brokers Kafka como middleware de comunicação (como discutimos e exercitamos em sala de aula).

Vocês poderão escolher o domínio de interesse. Deem prioridade a fontes de dados reais, disponíveis por meio de API's públicas. Alguns exemplos:

- API do Facebook: <https://developers.facebook.com/docs/graph-api>
- API do OpenSky Network, com dados de voos em tempo real: <https://openskynetwork.github.io/opensky-api/index.html#>
- API's de monitoramento de aspectos diferentes de redes: <https://bgpstream.caida.org/> e <https://search.censys.io/api>
- E tantas outras: <https://rapidapi.com/hub>

Caso não queiram usar dados reais ou escolham um domínio que não tenha API aberta disponível, vão precisar simular dados em tempo real, para alimentar o sistema de monitoramento.

### 3. FUNCIONALIDADES

O sistema deve monitorar continuamente eventos produzidos por produtores, em tempo real, e tomar ações diante de situações de interesse. Implementem:

- No mínimo, três situações de interesse no seu domínio, detectadas ao consumir “eventos primitivos” do Kafka;
- Além disto, pensem em uma situação um pouco mais complexa, na qual um consumidor processa “eventos primitivos”, infere conhecimento e alimenta novamente o Kafka com “eventos compostos ou derivados” (que podem ser consumidos para tomada de decisões/ações). Neste caso, o consumidor também exibe o papel de produtor (de eventos derivados).

Por exemplo, considere o domínio de monitoramento de temperatura de vacinas; as seguintes situações de interesse (e as respectivas ações) poderiam ser definidas:

- Caso a temperatura de uma câmara de vacina (evento simples) chegue perto dos limites dos parâmetros, deve-se notificar os gestores;
- Caso a temperatura de uma câmara de vacina fique fora dos limites dos parâmetros definidos durante um certo intervalo de tempo (evento derivado), o sistema deve sugerir o descarte do produto.

Funcionalidades desejáveis (extra):

- As informações sobre situações de interesse podem ser visualizadas em interfaces gráficas, ou planilhas, ou arquivos de texto, etc.;

**Divirtam-se!!! 😊**