

Laboratório 6

Objetivo: Saber usar a API Java no acesso ao serviço *Google Pub/Sub*. Compreender as diferenças entre tópico e subscrição e as garantias de entrega de mensagens. Compreender os padrões *fan-out* e *work-queue*. Saber integrar o Pub/Sub com outros serviços.

Nos últimos anos os serviços reguladores de trânsito rodoviário têm apostado na instalação de radares de controlo de velocidade nas estradas e arruamentos das cidades.

O desafio deste laboratório, como se apresenta na figura 1, é implementar um sistema distribuído com utilização de serviços na cloud (Google GCP) que permita receber os eventos gerados nos múltiplos radares, registá-los numa base de dados NoSQL e mais tarde emitir penalidades (multas) aos condutores prevaricadores de excesso de velocidade.

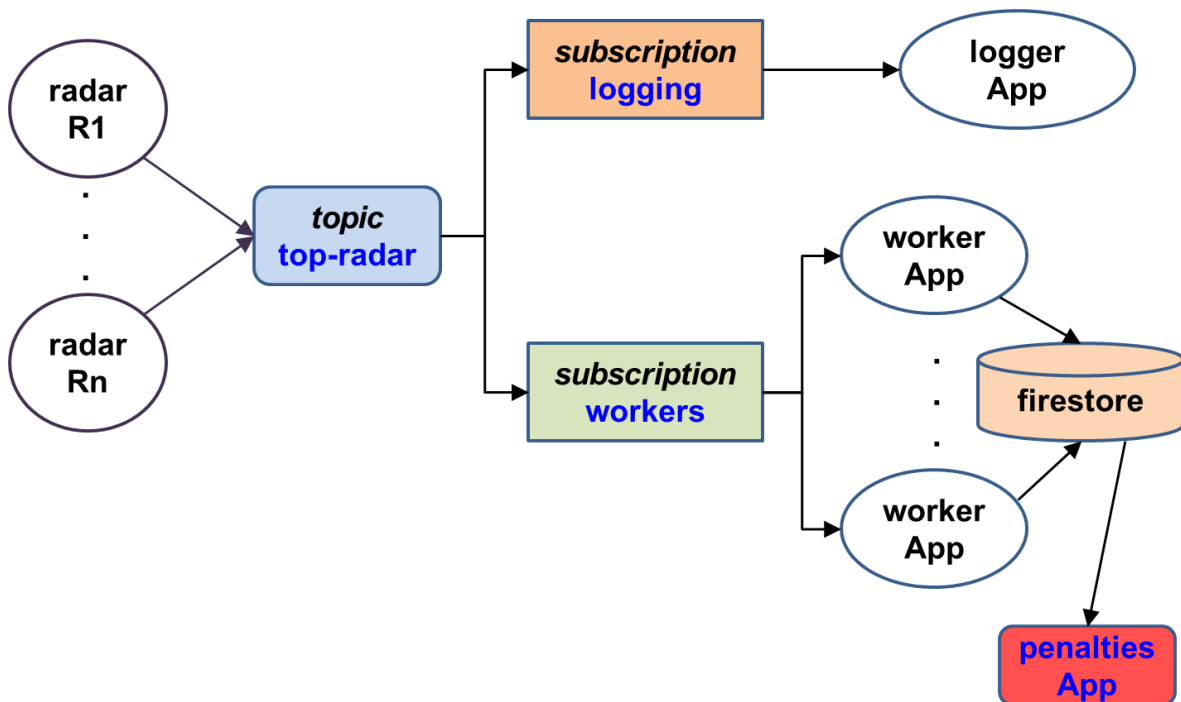


Figura 1 - Diagrama geral do sistema a implementar

- A. Usando a consola GCP, e tendo como referência a Figura 1, realize as seguintes ações:
 - a. Crie um tópico de nome *top-radar*;
 - b. Crie as subscrições *logging* e *workers* associadas ao tópico *top-radar*;
 - c. Usando a consola GCP, ou a aplicação exemplo apresentada nas aulas e publicada no Moodle, publique mensagens no tópico e verifique que são recebidas nas duas subscrições.
- B. Implemente um sistema de acordo com a Figura 1, que integre o serviço Pub/Sub e o serviço Firestore, composto pelas aplicações (radar, logger App, worker App e penalties App).

- A aplicação *radar App* envia mensagens, com eventos de excesso de velocidade, para o tópico (*top-radar*). Cada mensagem deve conter a matrícula do veículo e a velocidade detetada no formato: **XX-YY-ZZ:velocidade** e os seguintes atributos:
 - Local do registo do evento;
 - Data e hora;
 - Número de quilómetros acima do máximo permitido no local (*K_Acima*).
- Existem duas subscrição associadas ao tópico (*top-radar*):
 - i) Na subscrição *logging* existe um *subscriber* (*logger App*) que recebe todas as mensagens e limita-se a fazer *logging* (para simplificar escrevendo as mensagens no standard output);
 - ii) Na subscrição *workers* existem vários *workers* (*worker App*) que recebem mensagens em paralelo, criando no *firestore* documentos com a seguinte estrutura:
 - Um identificador de documento gerado pelo *firestore*;
 - Todos os dados e atributos da mensagem processada;
 - Um campo indicando o tipo de infração: (simples: $K_Acima < 20$; grave: $20 \leq K_Acima < 30$; muito grave: $K_Acima \geq 30$)
- A aplicação *penalties App* permite realizar queries à base dados, por exemplo:
 - Quais as infrações de um veículo;
 - Quais os veículos com infrações, simples, graves ou muito graves
 - etc.