| ISEL INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA<br>18/06/2024 | Departamento de Engenharia Eletrónica e Telecomunicações e de Computadores<br>**Computação na Nuvem (Época Normal)**<br>**LEIRT, LEIC, LEIM** |
|---|---|

**EXAME DE ÉPOCA NORMAL (2ª PARTE, COM CONSULTA, DURAÇÃO: 75 min., 9 valores)**

- ★ ***Nas questões 9 e 10, onde tem de apresentar código Java, não precisa de indicar o pom.xml, as declarações de import ou detalhes que não sejam essenciais (por exemplo static main). Utilize um misto de código Java e comentários para indicar eventuais pressupostos;***
- ★ ***A resposta a cada uma das questões (9, 10 e 11) tem de ser entregue em folhas separadas.***

**9) [3 val.]** Considere um cenário de uma fábrica onde se pretende controlar múltiplas máquinas através de um sistema Cliente/Servidor implementado em gRPC. Cada máquina é um cliente e o servidor é uma aplicação que recebe continuamente mensagens com medidas de temperatura das várias máquinas.

Consoante o valor de temperatura de cada máquina, o servidor envia comandos para que a máquina se configure com a seguinte lógica: i) Se o valor for abaixo de 100 não é necessário enviar comando; ii) Se o valor for entre 100 e 500 o servidor envia uma sequência de 3 comandos "`Cmd1`", "`Cmd2`" e "`Cmd3`"; ii) Se o valor for maior que 500 então o servidor envia à máquina o comando "`TurnOff`".

Considere o seguinte contrato *protobuf*:

```
service Fabric {

  rpc controller(stream Measure) returns (stream Command);

}
```

```
message Measure {
  string machID=1;
  double value=2;
}
message Command {
  string cmd=1;
}
```

a) **[1,5 val.]** Apresente o esboço do código Java que implementa a aplicação servidora, com especial ênfase na classe que implementa o contrato. Assuma que o servidor não armazena os valores de temperatura enviados pelas máquinas.

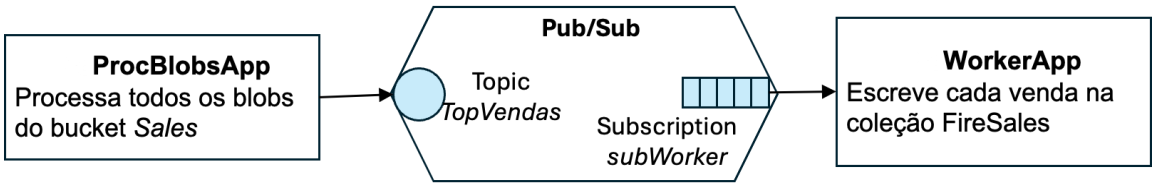
b) **[1,5 val.]** Apresente o esboço do código Java da aplicação cliente da máquina com ID="Mach99" que envia uma sequência de valores de temperatura: 10, 30, 150, . . ., 550, escrevendo na consola os comandos recebidos do servidor.

**(Questões 10 e 11 no verso)**

★ ***As respostas às questões 10 e 11 têm de ser entregue em folhas separadas***

**10) [4 val.]** Considere que uma empresa de vendas de produtos armazenou no GCP Storage, num *bucket* de nome *Sales*, milhares de *blobs* que são ficheiros CSV (*Comma Separated Value*) com as vendas de produtos realizadas ao longo dos anos. Cada ficheiro tem unicamente 1 venda com o seguinte formato: `Date, ProductID, ProductInfo, Quantity, UnitPrice, TotalPrice`.

Por forma a facilitar no futuro consultas aos dados, a empresa decidiu implementar um sistema como se descreve no diagrama da figura, o qual envolve uma aplicação para processar os *blobs*, o serviço Pub/Sub e uma aplicação para escrever os dados numa base de dados Firestore.




a) **[2 val.]** Apresente o esboço do código Java da aplicação *ProcBlobsApp* que processa todos os *blobs*, indicando também os pressupostos que ache necessários para executar a aplicação, nomeadamente as permissões de acesso aos serviços GCP envolvidos.

b) **[2 val.]** Apresente o esboço do código Java da aplicação *WorkerApp*, que processa as mensagens do Pub/Sub com as vendas. Cada venda fica armazenada como um documento na coleção `FireSales`. Indique também os pressupostos que ache necessários para executar a aplicação *worker* com possibilidade de elasticidade, bem como as permissões de acesso aos serviços GCP envolvidos.

**11) [2 val.] [Não é necessário apresentar código]**

No Trabalho Final existia um ou mais servidores gRPC intermediários entre a aplicação cliente e os serviços GCP, com a vantagem de não ser necessário cada aplicação cliente ter uma chave de uma conta de serviço com permissões de acesso aos recursos do projeto GCP.

Considere um cenário onde já existem muitas imagens JPG previamente armazenadas no serviço GCP Storage, num *bucket* de nome `myPhotos`. Dado que uma estratégia *serverless computing* com Google Cloud Functions oferece vantagens em termos de custos e de escalabilidade, proponha uma solução com funcionalidades idênticas às do Trabalho Final para processar todas as imagens já existentes no *bucket* `myFotos`, apresentando um diagrama de arquitetura e respetiva descrição, que maximize o uso de Cloud Functions.