

 ISEL <small>INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA</small>	Departamento de Engenharia Eletrónica e Telecomunicações e de Computadores Computação na Nuvem (Época Normal) LEIRT, LEIC, LEIM	
Nº	Nome:	22/06/2023

EXAME DE ÉPOCA NORMAL (1ª PARTE, SEM CONSULTA, DURAÇÃO: 45 min., 11 valores)

Nas questões 1 a 5 assinale as afirmações como verdadeira (V), falsa (F) ou não responde (). Uma opção assinalada corretamente soma 0,25 valores e incorretamente desconta 0,125 valores (50% da cotação de cada alínea), garantindo-se que uma questão nunca terá cotação menor que zero.

1. [1 val.] Sobre as características dos sistemas distribuídos

<input type="checkbox"/>	O teorema CAP demonstra que a escalabilidade de um sistema depende da capacidade de CPU
<input type="checkbox"/>	O desacoplamento (<i>loosely coupled</i>), entre as partes, depende da existência de contratos, por exemplo contratos <i>protobuf</i> em gRPC, para separar as implementações do cliente e do servidor
<input type="checkbox"/>	Um <i>stub</i> de cliente em gRPC, bloqueante ou não bloqueante, facilita a total transparência à localização que é uma das características desejáveis nos sistemas distribuídos
<input type="checkbox"/>	A Latência é um fator irrelevante, podendo ser zero em aplicações distribuídas

2. [1 val.] Relativamente ao serviço Storage do GCP:

<input type="checkbox"/>	As possíveis classes de acesso aos dados são, desde a maior para a menor frequência de acesso, respetivamente, <i>Standard</i> , <i>Nearline</i> , <i>Coldline</i> e <i>Archive</i> ,
<input type="checkbox"/>	Embora seja possível ter nomes compostos que dão ilusão de diretorias, o espaço de nomes de armazenamento de <i>blobs</i> no serviço de <i>Storage</i> segue um modelo <i>flat</i> (não hierárquico)
<input type="checkbox"/>	O <i>download/upload</i> do conteúdo de um <i>blob</i> através da API Java estudada só permite a transferência dos dados de uma só vez num único bloco de <i>bytes</i>
<input type="checkbox"/>	Os projetos <i>CN-2223</i> e <i>CN-2223-Alunos</i> podem cada um conter um <i>bucket</i> chamado <i>imagens-cn-2223</i> pois são projetos distintos

3. [1 val.] Relativamente ao serviço Firestore do GCP:

<input type="checkbox"/>	Dois projetos diferentes podem criar coleções com o mesmo nome
<input type="checkbox"/>	Dentro de uma coleção, todos os documentos têm de conter os mesmos campos
<input type="checkbox"/>	Uma interrogação sobre um único campo de documentos de uma coleção necessita de um índice previamente construído pelo programador
<input type="checkbox"/>	Pela API, podemos inserir documentos no <i>Firestore</i> a partir de objetos Java

4. [1 val.] Relativamente ao desenvolvimento de aplicações cliente/servidor em gRPC:

<input type="checkbox"/>	Numa operação de <i>stream</i> de servidor, as mensagens podem ser recebidas pelo cliente numa ordem diferente daquela que o servidor enviou
<input type="checkbox"/>	Na definição do contrato no <i>protobuf</i> fica indicado o tipo de <i>stub</i> que pode ser usado na invocação
<input type="checkbox"/>	Na operação <code>rpc oper(stream Req) returns (Void)</code> a implementação no servidor tem a assinatura <code>StreamObserver<Req> req(StreamObserver<Void> response)</code>
<input type="checkbox"/>	Na definição de uma operação num contrato <i>protobuf</i> só pode haver um tipo de mensagem de entrada e um tipo de mensagem de saída

5. [1 val.] Relativamente ao serviço Compute Engine do GCP (VM e Cloud Functions):

<input type="checkbox"/>	Uma VM ligada (<i>Started</i>) só gasta dinheiro se receber pedidos (ex: nos portos 80 ou 443)
<input type="checkbox"/>	Numa <i>Cloud Function</i> a assinatura do método (<i>entrypoint</i>) varia consoante o tipo de <i>trigger</i>
<input type="checkbox"/>	Uma <i>Cloud Function</i> com <i>trigger</i> HTTP pode ser chamada através de um pedido de HTTP GET
<input type="checkbox"/>	Ter uma aplicação em execução numa VM (com ligações HTTP) é em tudo equivalente a ter a mesma funcionalidade numa <i>Cloud Function</i> com <i>trigger</i> HTTP

6. [2 val.] Descreva de que forma o serviço Compute Engine do GCP permite implementar um servidor com propriedades de elasticidade.

7. [2 val.] Considere um sistema que tem por base o serviço Pub/Sub do GCP e um tópico para publicação de mensagens onde se pretende que todos os consumidores recebam todas as mensagens. Compare, com a ajuda de diagramas, a utilização dos padrões *fan-out*, *work-queue* ou a utilização de *Cloud Functions* com *trigger* Pub/Sub para a receção das mensagens.

8. [2 val.] Apresente o essencial do contrato *protobuf* de um servidor gRPC que implementa uma operação gRPC de nome **exame** como se indica a seguir:

```
public StreamObserver<Query> exame(StreamObserver<Result> responseObserver) {  
    return new StreamObserverQueries(responseObserver);  
}  
public class StreamObserverQueries implements StreamObserver<Query> {  
    StreamObserver<Result> result;  
    public StreamObserverQueries(StreamObserver<Result> result) { this.result=result; }  
    @Override  
    public void onNext(Query query) {  
        if (query.getNum() % 2 == 0)  
            result.onNext(Result.newBuilder().setIsEven(true).build());  
    }  
    @Override  
    public void onError(Throwable throwable) { }  
    @Override  
    public void onCompleted() { result.onCompleted(); }  
}
```

Operação do contrato:

Mensagens envolvidas na operação: