

 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:	5/07/2022

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.] Relativamente ao serviço Storage do GCP:

<input type="checkbox"/>	É possível adicionar uma imagem como Blob de nome <i>img.jpg</i> a um <i>Bucket</i> onde já existia um <i>Blob</i> com o mesmo nome e manter ambos os <i>Blobs</i> , desde que o <i>upload</i> seja feito usando contas de serviço diferentes.
<input type="checkbox"/>	Todas as aplicações que façam acessos a <i>Blobs</i> de um <i>bucket</i> de um determinado projeto GCP têm forçosamente de partilhar chaves da mesma conta de serviço.
<input type="checkbox"/>	Uma aplicação quando escreve um <i>Blob</i> num <i>Bucket</i> multiregional não pode escrever em cada região um conteúdo diferente do <i>Blob</i> .
<input type="checkbox"/>	Os <i>Blobs</i> com controlo de acesso público podem ser lidos por qualquer aplicação sem necessidade de a mesma ter uma chave associada a uma conta de serviço.

2. [1 val.] Ao desenvolver sistemas distribuídos deve ter-se em conta o desacoplamento entre as partes. Por exemplo, para ter desacoplamento numa aplicação cliente/servidor deve-se:

<input type="checkbox"/>	Usar no servidor <i>threads</i> diferentes para atender os múltiplos pedidos.
<input type="checkbox"/>	Definir um contrato (JAR com interface em Java RMI ou os <i>stubs</i> gerados a partir de contrato <i>protobuf</i>) para separar as implementações do cliente e do servidor.
<input type="checkbox"/>	Garantir que a implementação do servidor é <i>stateless</i> por forma a que o cliente se possa conectar a múltiplas instâncias do servidor permitindo assim distribuição de carga por escalabilidade horizontal (<i>scale out</i>).
<input type="checkbox"/>	Garantir que tanto o código do cliente como o código do servidor não partilham implementações de classes, para além das resultantes da definição do contrato e eventualmente de bibliotecas externas.

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

<input type="checkbox"/>	As aplicações podem fazer pesquisas usando uma linguagem de <i>script</i> semelhante à linguagem SQL do modelo relacional.
<input type="checkbox"/>	Os documentos têm de ter um identificador único que é um campo escolhido pelo programador.
<input type="checkbox"/>	A abordagem NoSQL com Firestore pode minimizar um problema conhecido como <i>impedance mismatch</i> e associado ao modelo relacional.
<input type="checkbox"/>	Para fazer interrogações (<i>queries</i>) compostas que envolvam múltiplos campos é necessário criar previamente um índice.

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

<input type="checkbox"/>	Como a chamada às operações do servidor gRPC podem ser realizadas via <i>stubs</i> bloqueantes ou <i>stubs</i> não bloqueantes, o servidor tem de implementar métodos diferentes para os dois tipos de chamada.
<input type="checkbox"/>	Quando existe <i>stream</i> de cliente e <i>stream</i> de servidor o número de mensagens enviadas em cada <i>stream</i> pode ser em número diferente, de acordo com os requisitos da aplicação.
<input type="checkbox"/>	Num contrato <i>protobuf</i> podem existir operações que não são invocáveis com todos os tipos de <i>stub</i> .
<input type="checkbox"/>	A operação <code>rpc oper(stream Request) returns (Void)</code> pode ser chamada com <i>stub</i> bloqueante.

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

<input type="checkbox"/>	Para criar múltiplas VMs idênticas que se iniciam com a mesma configuração deve criar-se primeiro um <i>template</i> e depois criar as VMs a partir desse <i>template</i> .
<input type="checkbox"/>	É possível criar um <i>instance group</i> de VMs com <i>autoscale On</i> diretamente a partir da imagem de uma VM previamente criada e configurada.
<input type="checkbox"/>	As <i>Cloud Functions</i> com <i>trigger</i> HTTP devem ter código diferente para chamadas a partir de componentes alojados dentro ou fora da plataforma GCP.
<input type="checkbox"/>	O código de uma <i>Cloud Function</i> com <i>trigger</i> Pub/Sub apenas pode usar o serviço Pub/Sub, isto é, não é possível usar os restantes serviços GCP, por exemplo o Firestore.

6. [2 val.] Justifique porque é que no modelo *serverless*, nomeadamente nas *Cloud Functions* da plataforma GCP, o código a executar tem de ser *stateless*.

7. [2 val.] Considere um sistema onde vários componentes trocam mensagens através do serviço Pub/Sub. Alguns componentes enviam mensagens para um tópico e outros componentes processam essas mensagens num padrão *work-queue*. Após o sistema já estar em produção, verificou-se a necessidade de registar todas as mensagens para efeitos de *logging*. Proponha uma solução simples, sem alterar o que está em funcionamento, para poder implementar o mecanismo de *logging*.

8. [2 val.] Considere que é preciso desenvolver uma aplicação web relacionada com a gestão desportiva, onde existem os seguintes dados: clubes com vários jogadores, um jogador pode ter estado em vários clubes e existem dados sobre eventos desportivos, envolvendo clubes e jogadores, com uma estrutura muito diversa (*schemaless*). Justifique se optava unicamente por uma base de dados relacional, unicamente por uma base de dados NoSQL ou um sistema híbrido.