

 ISEL <small>INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA</small>	Departamento de Engenharia Eletrónica e Telecomunicações e de Computadores Computação na Nuvem (Época de Recurso) LEIRT, LEIC, LEIM	
Nº	Nome:	21/07/2022

EXAME DE ÉPOCA RECURSO (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 chamadas remotas com os *middleware* Java RMI e gRPC:

<input type="checkbox"/>	Ambas permitem desacoplamento forte na medida em que as implementações das aplicações cliente e servidor só dependem de contratos
<input type="checkbox"/>	Quando um cliente envia um pedido ao servidor, em RMI o servidor pode enviar várias respostas de <i>Callback</i> . No entanto, em gRPC não é possível ao servidor enviar várias respostas a um pedido.
<input type="checkbox"/>	O gRPC é mais genérico que o RMI pois usa um contrato neutral (<i>protobuf</i>) que permite gerar <i>stubs</i> em múltiplas linguagens de programação. Contrariamente o RMI está dependente da linguagem Java.
<input type="checkbox"/>	Tanto em RMI como em gRPC o acesso do cliente a todas as operações do servidor pode ser feito com <i>stubs</i> bloqueantes ou <i>stubs</i> não bloqueantes

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

<input type="checkbox"/>	Não pode haver <i>buckets</i> com o mesmo nome em regiões diferentes
<input type="checkbox"/>	Um <i>bucket</i> multi-regional permite diminuir a latência de acesso aos <i>blobs</i> a partir de programas a executar nessas regiões
<input type="checkbox"/>	Para aceder a um <i>blob</i> público precisamos de uma conta de serviço com permissões de acesso público
<input type="checkbox"/>	Dentro do mesmo <i>bucket</i> todos os <i>blobs</i> têm de ter os mesmos metadados

3. [1 val.] Relativamente aos serviços GCP e contas de serviço:

<input type="checkbox"/>	Enquanto o serviço <i>Firestore</i> é específico de um projeto GCP, o serviço <i>Storage</i> é partilhado por todos os projetos GCP, implicando que o nome de um <i>bucket</i> tem de ser único globalmente na plataforma GCP
<input type="checkbox"/>	Uma conta de serviço para dar permissão de acesso aos recursos está ligada a um projeto GCP, gastando créditos na conta de faturação associada ao projeto
<input type="checkbox"/>	Uma VM com o porto 8000 aberto, com uma regra de <i>firewall</i> , existe no contexto de um projeto GCP, mas pode ser acedida através desse porto sem configuração de uma conta de serviço
<input type="checkbox"/>	Uma <i>Cloud Function trigger</i> HTTP pode utilizar todos os recursos com as permissões definidas na conta de serviço indicada no <i>deployment</i> da função

4. [1 val.] Sobre os conceitos de Escalabilidade e Elasticidade:

<input type="checkbox"/>	Escalabilidade e Elasticidade têm o mesmo significado
<input type="checkbox"/>	Elasticidade é a capacidade de efetuar escalabilidade automática (aumentar ou diminuir os recursos)
<input type="checkbox"/>	Perante os picos de trabalho de um sistema, por exemplo número de clientes, se aumentarmos o número de máquinas para balanceamento de carga, então também podemos dizer que o sistema tem a propriedade de Elasticidade
<input type="checkbox"/>	Os <i>instance group</i> de VMs da plataforma GCP permitem implementar facilmente Elasticidade

5. [1 val.] Considere um servidor que disponibiliza a seguinte operação gRPC:
`rpc oper(stream ZipBlock) returns (Descriptor);`

<input type="checkbox"/>	Após a chamada da operação, o cliente pode enviar uma sequência de mensagens <i>ZipBlock</i> e no final recebe uma única mensagem do tipo <i>Descriptor</i>
<input type="checkbox"/>	A assinatura do método que implementa a operação no servidor é: <code>StreamObserver<Descriptor> oper(StreamObserver<ZipBlock>)</code>
<input type="checkbox"/>	A assinatura do método que implementa a operação no servidor é: <code>StreamObserver<ZipBlock> oper(StreamObserver<Descriptor>)</code>
<input type="checkbox"/>	Na aplicação cliente, a operação <i>oper</i> tem forçosamente de ser chamada usando um <i>stub</i> não bloqueante

6. **[2 val.]** Indique pelo menos duas diferenças entre ter uma *Cloud Function trigger* HTTP e ter o código equivalente num servidor HTTP numa máquina virtual no GCP?

7. **[2 val.]** Considere que durante as férias tirou fotos magníficas em grande quantidade e qualidade e que quer partilhar com os seus amigos e familiares. Infelizmente chegou à conclusão de que o ficheiro zip com todas as fotos não é passível de ser enviado como anexo num email. Descreva como poderia ultrapassar a situação tirando partido do serviço *Storage* do seu projeto GCP.

8. **[2 val.]** Considere um sistema de um jogo de apostas, em que múltiplos utilizadores enviam mensagens com uma aposta (conjunto de 5 números). Essas apostas, em grande quantidade, têm de ser processadas rapidamente por um conjunto de servidores em que todos eles conhecem a chave dos 5 números com prémio. A chave e as apostas devem estar armazenadas de forma persistente, incluindo informação se a aposta tem ou não prémio. Descreva com ajuda de um diagrama a arquitetura do sistema baseada nos serviços GCP *Pub/Sub*, *Compute Engine* e *Firestore*. (note que não é necessário especificar o formato das mensagens e dos dados no *Firestore*)