

# Sistemas Distribuídos, 2018/19

## 1º MINI Teste

- Todas as perguntas têm a mesma cotação. Cada pergunta tem apenas uma resposta completamente certa.
- Na sua resposta pode selecionar uma ou mais alíneas. Preencha-as por ordem crescente, com vírgulas.
- Para cada pergunta, a nota é calculada pelas alíneas que escolheu na sua resposta, da seguinte forma: a alínea correta conta com a cotação completa; cada alínea incorreta desconta 1/3 da cotação da pergunta.
- Exemplo: numa dada pergunta, escolheu as alíneas "A, D". Se a alínea certa for a A, então a nota final será 2/3 da cotação (cotação completa pela alínea certa menos 1/3 pela alínea incorreta).

Número: \_\_\_\_\_ Nome: \_\_\_\_\_

1) Qual das seguintes semânticas RPC exige manter estado no servidor para suportar o protocolo pedido-resposta?

- A. Talvez
- B. Pelo menos uma vez
- C. No máximo uma vez
- D. Nenhuma.

2) As ferramentas do gRPC permitem:

- A. Implementar servidor e cliente em <sup>potencialmente</sup> qualquer linguagem de programação.
- B. Apenas implementar servidor e cliente na mesma linguagem de programação.
- C. Implementar servidor em Java, mas o cliente noutra linguagem.
- D. Implementar servidor e cliente apenas em Java. <sup>diferente</sup>

3) Considere o seguinte ficheiro .proto que vai ser usado para gRPC:

```
message HelloRequest {
  string name = 1;
}
message HelloResponse {
  string message = 1;
}
```

- A. Estas definições são suficientes para definir a operação remota *Hello*.
- B. Falta a definição *service* com especificação do *rpc Hello*
- C. O tipo *string* não existe e tem que ser substituído do *char\**
- D. O número 1 não pode ser usado nas diferentes mensagens.

4) Considere a seguinte interface Java:

```
public interface Chess extends java.rmi.Remote {
  public String getBoard() throws java.rmi.RemoteException;
}
```

Qual é a afirmação verdadeira?

- A. A interface não pode ser usada como está porque o método não atira *EmptyBoardException*.
- B. Em vez de *String*, os métodos apenas podem devolver *Remote*.
- C. Em vez de *String*, os métodos apenas podem devolver *Serializable*.
- D. A interface pode ser usada para definir objetos remotos tal como está.

- 5) Considere o seguinte extrato de um programa em Java RMI, em que X é uma interface remota mas Y não é. Que proxies são instanciados neste programa?

```
X x = (X) Naming.lookup(name);  
Y y = x.m(1);  
y.d(1000);
```

- A. Nenhum.
- B. Um proxy, referenciado pela variável x.
- C. Um proxy, referenciado pela variável y.
- D. Dois proxies, referenciados pelas variáveis x e y.

- 6) Considere o seguinte extrato de um programa em Java RMI, em que AccountList é interface remota mas Account não é. Considerando apenas as linhas 2 e 3, em qual/quais pode originar uma RemoteException?

```
1 AccountList acList = (AccountList) Naming.lookup(remoteObjectName);  
2 Account a = acList.getAccount(1);  
3 a.debit(1000);
```

- A. Nenhuma.
- B. Linha 2.
- C. Linha 3.
- D. Ambas (2 e 3).

- 7) O que é o WSDL?

- A. Um protocolo de comunicação para *Web Services*.
- B. Um contrato que descreve de forma rigorosa as operações disponibilizadas por um serviço.
- C. Uma descrição de tipos de dados.
- D. Um contrato que define as obrigações legais de um prestador de serviço.

- 8) Na arquitetura de *Web Services*, o protocolo de transporte suportado é:

- A. HTTP, SMTP e outros
- B. Apenas HTTP
- C. Apenas Filas de mensagens (*message queues*)
- D. HTTP e outros, mas não SMTP

- 9) Considere a seguinte mensagem SOAP:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">  
  <S:Body><ns2:sayHello xmlns:ns2="http://ws.example.com/">  
    <arg0>friend</arg0>  
  </ns2:sayHello></S:Body></S:Envelope>
```

- A. A mensagem está de acordo com a norma.
- B. A mensagem não está de acordo com a norma porque falta o <header>
- C. A mensagem não está de acordo com a norma porque falta a referência para o WSDL.
- D. A mensagem não está de acordo com a norma porque a regra de indentação não foi respeitada.