

# 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 é a diferença principal entre uma chamada remota síncrona e assíncrona?
- A. A chamada síncrona permite que a função remota se execute mais cedo, ao contrário da assíncrona.
  - B. A chamada síncrona é bloqueante, a assíncrona não.
  - C. A chamada assíncrona é bloqueante, a síncrona não.
  - D. O mesmo cliente (*single-thread*) apenas pode fazer uma chamada assíncrona de cada vez.

- 2) Na arquitetura gRPC, o protocolo de transporte suportado é:
- A. HTTP nas versões 1.1 e 2.0
  - B. HTTP/2, SMTP e outros
  - C. Apenas filas de mensagens (*message queues*) e SMTP
  - D. Apenas HTTP/2

- 3) Considere o seguinte ficheiro .proto que vai ser processado pela ferramenta protoc com linguagem alvo Java:

```
service HelloWorld {
  rpc SayHello (HelloRequest) returns (HelloResponse) {}
}
message HelloRequest {
  string name = 1;
}
message HelloResponse {
  string message = 1;
}
```

A assinatura da operação SayHello em Java vai ser:

- A. `public void sayHello(HelloRequest request, StreamObserver<HelloResponse> responseObserver)`
- B. `public String sayHello(String name)`
- C. `public void sayHello(String name, String message)`
- D. `public String sayHello(String name) throws RemoteException`

- 4) Considere a seguinte interface Java:

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

Qual é a afirmação verdadeira?

- A. A interface não pode ser usada como está porque o método não atira `java.rmi.RemoteException`.
- 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 e Y são interfaces remotas. 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) Uma grande diferença entre os modelos de programação por RPC e por RMI é que:

- A. RMI suporta passagem de objetos por referência, ao contrário do RPC.
- B. A implementação das rotinas remotas no servidor pode recorrer a uma linguagem orientada a objetos em RMI, ao contrário do RPC.
- C. O RMI é concebido de raiz para ser usado entre nós heterogêneos na internet, ao contrário do RPC.
- D. O RMI é, por natureza, mais tolerante a falhas que o RPC.

7) Um documento XML considera-se válido se:

- A. For bem-formatado, isto é, cumpre as regras de sintaxe do XML.
- B. For bem-formatado e cumprir a especificação definida por um *XML Schema Definition* (XSD)
- C. For bem-formatado e os dados nele contidos forem corretos.
- D. For bem-formatado e o seu tamanho total inferior a 100 Kilobytes.

8) A implementação dos Web Services em Java, JAX-WS, permite:

- A. Fazer clientes que comunicam com serviços implementados noutras linguagens.
- B. Fazer clientes que comunicam apenas com serviços implementados na mesma linguagem de programação.
- C. Intercetar as mensagens SOAP e ler ou modificar o seu conteúdo.
- D. A e C

9) Qual a semântica de execução de uma invocação remota com SOAP/HTTP?

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