

LETI/LEIC 2015-2016, 1º Exame de Sistemas Distribuídos**14 de junho de 2016**

Responda no enunciado, usando apenas o espaço fornecido. Identifique todas as folhas.

Uma resposta errada numa escolha múltipla desconta $1/(N-1)$ do valor da pergunta (sendo N alternativas).

Duração da prova: 2h30m

Grupo I [3 valores]

```
import java.rmi.*;
public interface BankAccount extends Remote {
    public boolean deposit(float amount) throws RemoteException;
    public boolean withdraw(float amount) throws
        OverdrawnException, RemoteException;
    public String getBalance() throws RemoteException;
}
```

- 1) [0,5] Considere a interface acima. Procure escrevê-la na IDL do SUN-RPC, inclua todos os elementos que achar necessários propondo os que não conseguir obter diretamente da interface acima. Não se preocupe nesta fase com as exceções.

--

- 2) [0,3] A interface em Sun-RPC é semelhante ao código C que é frequente aparecer em ficheiros .h. Contudo na alínea anterior deverá ter usado aspetos que diferenciam a interface em Sun-RPC de um interface em C. Explique quais e porquê.

- 3) [0,3] Considerando agora as exceções do método withdraw(). Que solução deveria adotar para conseguir que o cliente deste procedimento em Sun-RPC tivesse a mesma informação que tem o método remoto em Java.

4) As mensagens de Sun-RPC vão transmitir os dados de forma a garantir que estes sejam corretamente utilizados em qualquer tipo de máquina.

a) [0,2] Explique a razão deste problema ilustrando com um exemplo da interface.

b) [0,3] Que tipo de políticas usa o protocolo do Sun-RPC para tratar este problema. Justifique.

5) [0,4] Considerando as operações cuja interface se encontra acima quais consideraria como idempotentes. Justifique.

6) [0,4] Que tipo de semântica de invocação teria de ser usada para as outras operações. Justifique.

7) Suponha que quer utilizar como protocolo de transporte o UDP. Indique claramente que mecanismos seriam necessários para conseguir esta semântica.

a) [0,3] No cliente.

b) [0,3] No servidor.

Grupo II [3,5 valores]

Considere o excerto de um programa que procura criar um sistema muito simples de leilões utilizando Java RMI. Para tornar mais claro o programa e concretizar a topologia em que o mesmo está a ser usado considere a Figura 1 que deve ter em conta nas respostas seguintes.

```
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public interface Bid extends Serializable {
    int getAmount();
    int getBidderID();
}
public class BidImpl implements Bid { ... }

public interface BidListener extends Remote {
    void tellMeABid(Bid newBid) throws RemoteException;
}
public class BidListenerImpl implements BidListener { ... }

// Runs an auction, informing all listeners of all bids.
public interface Auction extends Remote {
    void registerBL(BidListener bl) throws RemoteException;
    void makeBid(Bid newBid) throws RemoteException;
}

public class AuctionI extends UnicastRemoteObject implements Auction {
    public void registerBL(BidListener bl) throws RemoteException
    // Add bl to a collection bls }
    public void makeBid(Bid newBid) throws RemoteException {
    // for each bl in collection bls ...bl.tellMeABid(newBid);
    }
}
```

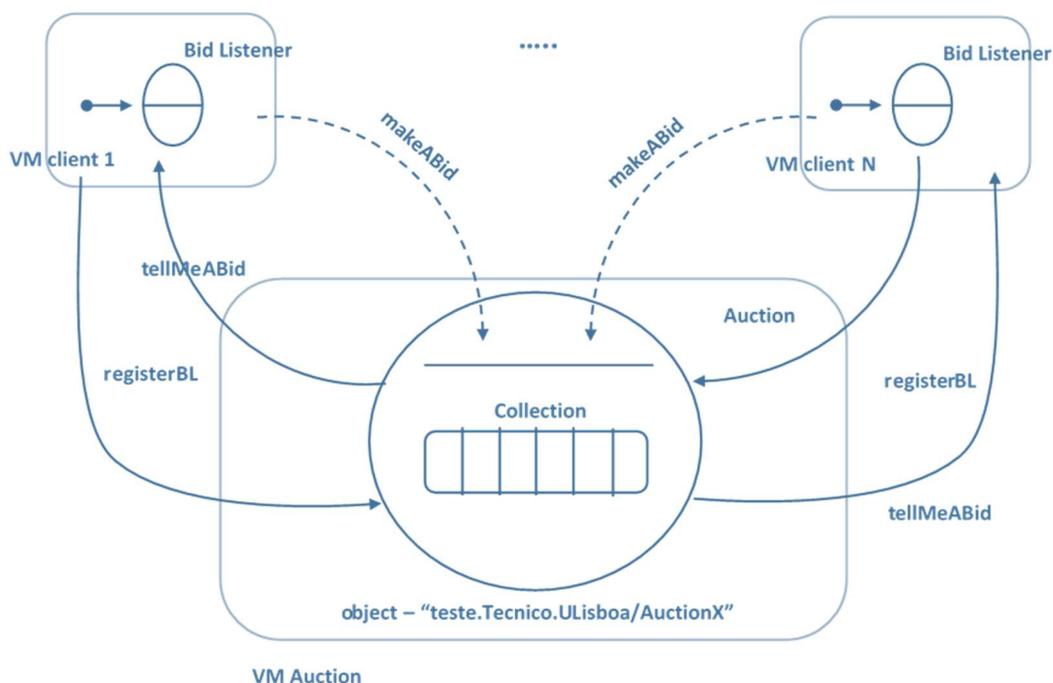


Figura 1

1) Suponha que o cliente no seu programa executa o seguinte conjunto de instruções

```
blClient1 = new BidListenerImpl();  
auction.registerBl(blClient1);
```

a) [0,5] Para conseguir executar estas instruções o que necessita o cliente de fazer anteriormente. Programe o que achar necessário.

b) [0,4] Suponha que este cliente quer fazer um novo lance no leilão. Programe como o faria.

2) O Client1 efectuou new do objeto BidListener

a) [0,3] Explique que operações são executadas pelo RMI quando é efetuado o registo no objeto Auction.

b) [0,4] Quando Auction invoca tellMeABid poderiam existir diversos clientes registados. De que forma é que o objeto Auction consegue ter o contexto de comunicação para aceder as máquinas e sockets correctos. Justifique claramente.

c) [0,4] O método tellMeABID tem como parâmetro newBid. Explique como é efectuada a transferência deste parâmetro e porquê.

3) Considere que registerBL foi invocado já 6 vezes e makeBid 10.

a) [0,4] Quantos proxies estão presentes na JVM de auction?

b) [0,4] Quantos proxies na VM Cliente 1?

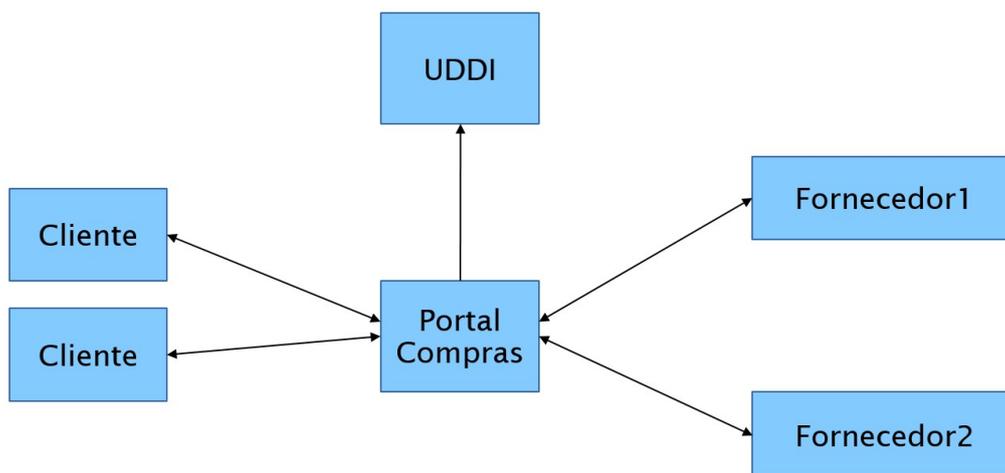
4) O Client1 termina mas não a JVM onde se executa

a) [0,3] O que sucede ao objeto BLClient1? Justifique.

b) [0,4] O que seria necessário para o sistema gerir corretamente a ciclo de vida dos objetos BidListener. Justifique.

Grupo III [3,5 valores]

Considere o seguinte sistema distribuído, baseado na tecnologia de Web Services. O portal de compras atende múltiplos clientes e efetua vendas *on-line*. O portal encaminha as pesquisas para os seus fornecedores.



1) Considere o componente 'UDDI' apresentado na figura.

a) [0,3] Qual é o seu papel no sistema?

b) [0,3] Qual é a informação mínima que deve constar do registo UDDI?

--

c) [0,4] Um registo UDDI pode ter também classificações. Indique um tipo de classificação de serviço e exemplifique a sua utilidade.

2) [0,5] Considere os 'Clientes' apresentados na figura. Cada cliente tem que ser construído na mesma linguagem de programação que os outros clientes? Justifique.

3) Suponha que quer interceptar as comunicações entre o 'Portal Compras' e os Fornecedores.

a) [0,6] Escreva, em pseudo-código, um SOAP Handler (interceptor) que imprime para a consola apenas as mensagens que saem do 'Portal Compras' (mas não imprime as que chegam).

--

b) Considere que tem acesso a uma biblioteca que contém os seguintes SOAP Handlers já desenvolvidos:

- *Logger* - imprime todas as mensagens SOAP para a consola
- *Zipper* - aplica o algoritmo de compressão/descompressão ZIP ao conteúdo da mensagem SOAP, guardando o conteúdo numa etiqueta chamada <zipBody> em vez de <body>

Considere a seguinte configuração de interceptores:

- Cliente: *Logger, Zipper, Logger*
- Servidor: *Zipper*

i) [0,4] Quando é feita a invocação remota de uma operação, por exemplo **order()**, qual a sequência de execução, desde o cliente até que a resposta seja recebida?

Seq. Pedido:

Seq. Resposta:

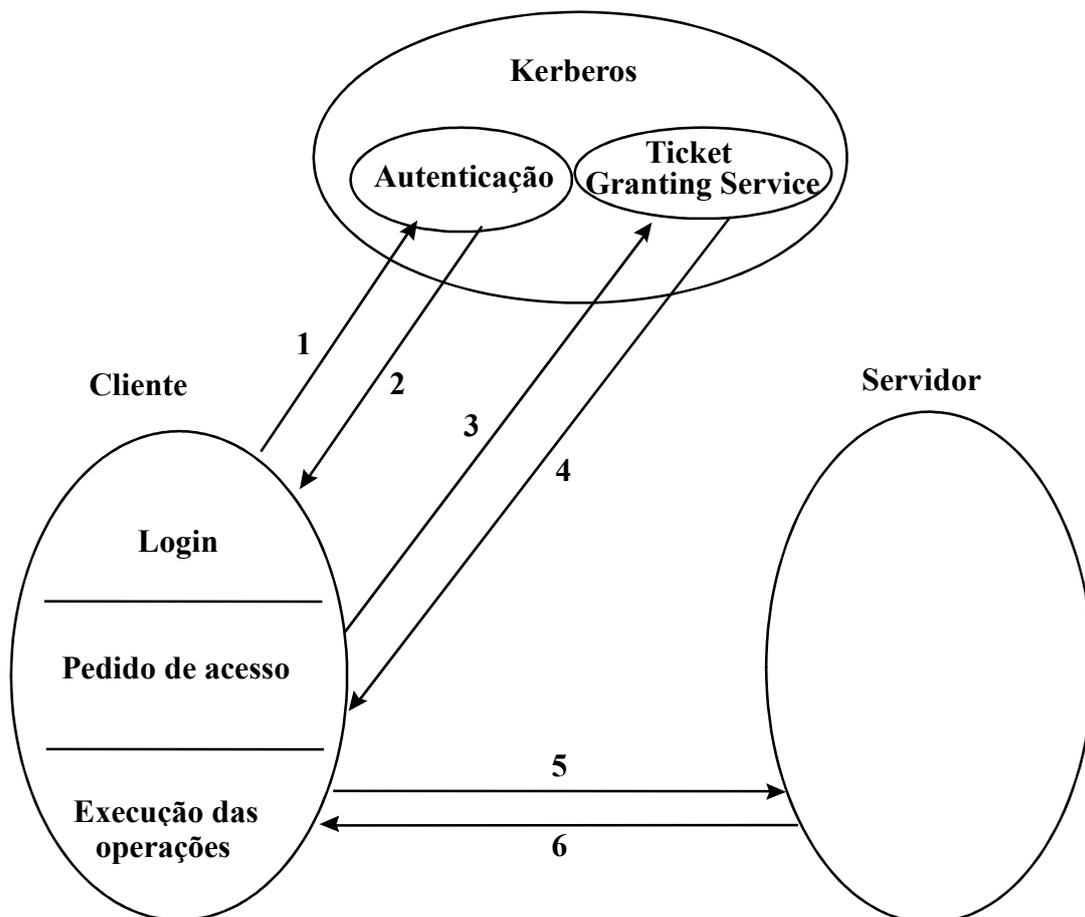
ii) [0,3] É possível ao cliente remover os *Loggers* sem recompilar/reconfigurar o servidor? Porquê?

iii) [0,3] É possível ao servidor remover o *Zipper* sem recompilar/reconfigurar o cliente? Porquê?

iv) [0,4] Considera que a utilização do intercetor *Zipper* no servidor quebra o contrato WSDL? Se sim, indique o que acrescentaria ao WSDL. Se não, indique o elemento WSDL relevante.

Grupo IV [4 valores]

1) Considere o seguinte diagrama que representa um sistema **Kerberos**.



a) [0,7] Complete a seguinte legenda para descrever o que acontece em cada passo do protocolo, evidenciando todos os artefactos relevantes:

1. O cliente identifica-se enviando um pedido de início de sessão para o SAut
2.
3.
4.
5.
6. O servidor envia o resultado da operação ao cliente, cifrado com Kcs

b) [0,4] Que chaves devem ser partilhadas antes do protocolo se poder executar?
Para cada chave referida, indique quem as conhece.

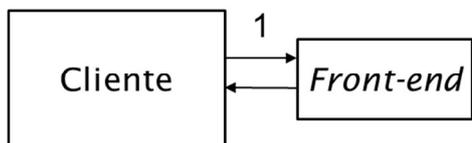
c) [0,4] Como pode o Servidor ter a certeza que o *ticket* que recebe é autêntico?

d) [0,4] Qual seria a vantagem de cifrar o *ticket* Kerberos com uma cifra por blocos em modo CBC (Cipher-Block Chaining) em vez de modo ECB (Electronic Code Book)? Justifique.

2) Considere novamente o cenário de um Portal de Compras do Grupo III, baseado na tecnologia de **Web Services**. Agora pretende-se garantir **segurança** para a comunicação entre o Portal e os Fornecedores, através de certificados digitais de chave pública seguindo a norma **X.509**.

a) [0,6] Em que consiste um certificado digital de chave pública?
Refira especificamente o que contém o certificado.

- a) [0,7] Considere que o cliente quer escrever na variável X o valor $22 - W(X,22)$.
 Complete o diagrama abaixo indicando as mensagens através de setas numeradas e acrescente a legenda necessária para cada número. Não represente as mensagens de "I'm alive".



R1

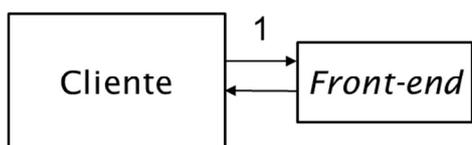
R2

R3

- b) [0,6] Qual será o tempo máximo de recuperação do sistema em caso de falha de R1?
 Use $t_{MaxEscolha}$ para representar o tempo máximo de escolha do novo primário.
 Defina outros parâmetros de que necessite.

- 2) Considere agora que o protocolo usado é o **quorum consensus** e que o sistema é assíncrono.
 O valor inicial da variável Y é 0.

- a) [0,7] Considere que o cliente quer executar a operação $W(Y, 6)$.
 Complete o diagrama com todas as mensagens numeradas e respetiva legenda.
 Considere que R1 não recebe a mensagem de escrita.

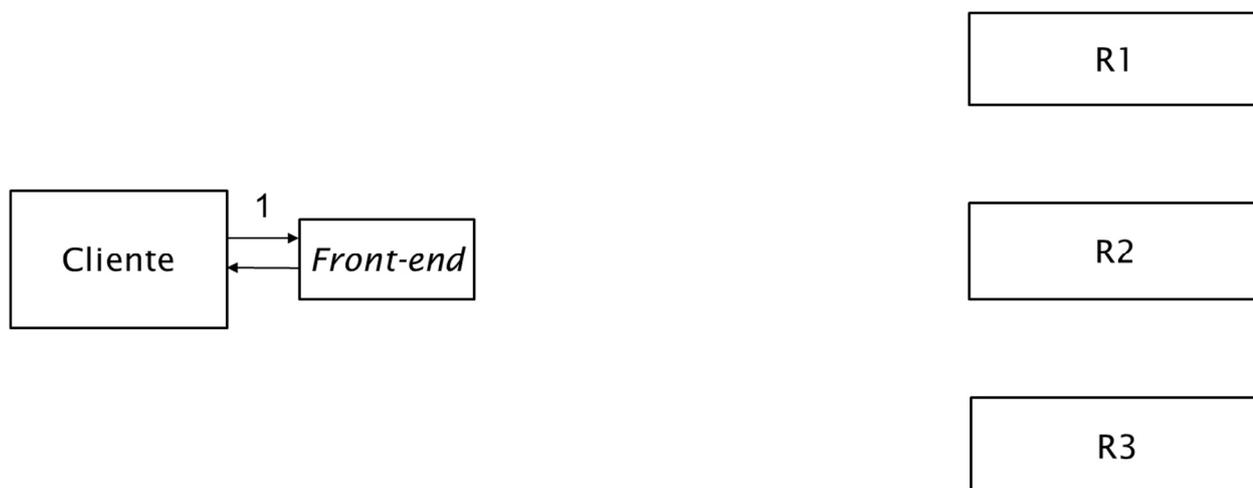


R1

R2

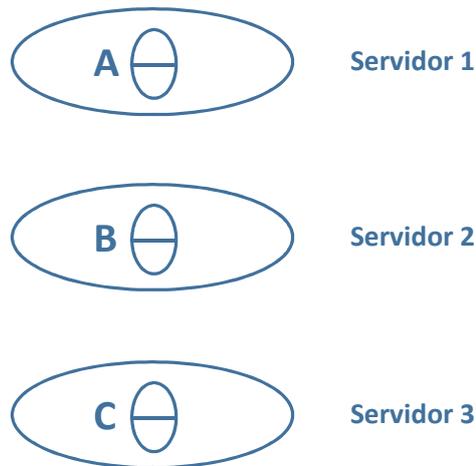
R3

- b) [0,5] Considere agora a operação $R(Y)$ que se executa posteriormente à escrita anterior. R1 está agora em cima, mas R3 está em baixo. Complete o diagrama.



- c) [0,5] Nesta situação, a leitura $R(Y)$ vai devolver o valor correto? Justifique como pode o cliente escolher corretamente entre o valor 0 e o valor 6.

Grupo VI [1,5 valores]



Considere um sistema distribuído ilustrado na figura acima e os programas de dois clientes que utilizam **transações distribuídas**. Assuma que os servidores usam um sistema transacional com política de sincronização pessimista. Os programas seguem uma sintaxe semelhante à do livro Coulouris da cadeira.

Cliente 1

```
T = OpenTransaction;
  Value = A.read();
  B.deposit(Value*0,05);
CloseTransaction;
```

Cliente 2

```
T = OpenTransaction;
  Value = C.read();
  B.deposit(Value*0,05);
CloseTransaction;
```

1) Considere o seguinte Diário (Log) no Coordenador:

<p>Cliente 1 Open TX Id = 271</p>	<p>Cliente 2 Open TX Id = 272</p>	<p>Servidor 3 Join 272</p>	<p>Servidor 1 Join 271</p>
--	--	---	---

a) [0,4] O que pode inferir que aconteceu até esse instante. Justifique.

- b) [0,2] O cliente 1 executa a operação remota de read.
 - i) A operação é executada sobre o coordenador e deverá ser registada no seu log.
 - ii) A operação já foi executada dando origem ao Join do servidor 1.
 - iii) A operação read obriga a fazer Join e o servidor informa o coordenador do resultado do read.
 - iv) O Join foi efetuado no início da transação por todos os servidores envolvidos.

- c) [0,2] Suponha que ambos os clientes continuam a executar os seus programas em paralelo e executam a instrução B.deposit(). O que sucede:
 - i) Não é possível porque apenas um pode aceder ao objeto B, uma das transações aborta.
 - ii) A transação 272 que começou depois da 271 bloqueia-se.
 - iii) Uma das transações bloqueia-se num trinco associado ao objeto B, não existindo qualquer situação de aborto.
 - iv) Nesta situação os servidores avisam o coordenador de um conflito no acesso.

- 2) Considere agora que o cliente 2 chegou à execução da operação CloseTransaction().
 - a) [0,4] Preencha o Log do coordenador desde o instante do CloseTransaction() até ao instante em que o coordenador pode tomar uma decisão de commit para finalizar a transação.
Use uma linha para cada Procure evidenciar todos os registos necessários.

- b) [0,3] O protocolo de *two-phase commit* tolera faltas temporárias da comunicação ou atrasos das mensagens na rede. Explique porquê tolera apenas faltas de **paragem temporárias da rede**.

Grupo VII [1,5 valores]

- 1) [0,3] Praticamente todos os sistemas cliente servidor têm associado um espaço de nomes e um gestor de nomes. Justifique a utilidade desta associação no caso do SUN-RPC. Procure mostrar que valor acrescenta ao DNS já existente.

2) [0,2] Considere o nome BANCOPROG presente neste extrato de uma especificação em Sun-RPC:

```
program BANCOPROG {  
  version BANCOVERS {  
    criarRet CRIAR(criarIn) = 1;  
    saldoRet SALDO(int) = 2;  
    resultado DEPOSITAR(contaEvalor) = 3;  
  } = 1;  
} = 0x20000005;
```

- A. Nome de um serviço local a um servidor, numa rede a executar Sun-RPC.
- B. Nome impuro.
- C. Não é um nome (no sentido informático) porque não é possível garantir a sua unicidade referencial.
- D. Nome de um serviço global a uma rede a executar Sun-RPC.

3) No DNS existe uma política de tolerância a faltas.

a) [0,4] Explique em detalhe qual é.

b) [0,4] Contudo o protocolo de replicação *primary- backup* não é utilizado no DNS. Explique qual a razão de não ser usado.

4) [0,2] No DNS compare o método de resolução iterativo e recursivo para otimizar o tempo de resposta dos servidores envolvidos a **futuros pedidos**:

- A. Iterativo é mais eficaz.
- B. Recursivo é mais eficaz.
- C. Não tem influência sobre a resolução de nomes no futuro.
- D. Iterativo e Recursivo são equivalentes neste aspeto.