LETI/LEIC 2015-2016, 2º Exame de Sistemas Distribuídos

28 de junho de 2016

Responda no enunciado, usando apenas o espaço fornecido. Identifique todas as folhas.
Uma resposta errada numa escolha múltipla com N opções desconta 1/(N-1) do valor da pergunta.
Duração da prova: 2h30m

Grupo I [3 valores]

Considere o seguinte programa em Java:

```
public class Car implements Serializable {
	private String make;
	private int mileage;
	// ...
}

public interface ICarManager extends Remote {

	// Looks for an available car that is parked within a specified radius of the
	// coordinates supplied as arguments;
	// if more than one car is available, the nearest is returned
	Car findNearestFreeCar(float latitudeCoordinates, float longitudeCoordinates,
		int maxRadiusMeters) throws RemoteException, NoAvailableCar;
...
```

1) Considere a interface acima, procure escrevê-la na IDL do SUN-RPC.
   a) [0,6] Em primeiro lugar, programe as estruturas de dados necessárias para os parâmetros dos RPC do SUN-RPC de modo a obter **exatamente o mesmo funcionamento** desta interface em Java.

```
struct                          struct
```

   b) [0,6] Complete agora a IDL Sun-RPC do serviço `findNearestFreeCar`. Inclua todos os elementos que achar necessários, propondo os que não conseguir obter diretamente da interface acima.

2) Considere que a semântica do RPC utilizada é "pelo-menos-uma-vez" (*at-least-once*). O cliente efetua a chamada a `findNearestFreeCar`. Indique no campo "Resultado para o cliente" se o RPC retorna um resultado válido ou inválido (`RPC_SUCCESS` ou `RPC_ERROR`, respetivamente) e indique no segundo campo o número de vezes que a função é executada no servidor.

   a) [0,4] A mensagem de invocação inicial é perdida

Resultado para o cliente	Número de vezes executada no servidor

   b) [0,4] A mensagem de resposta do servidor é perdida pela rede 2 vezes

Resultado para o cliente	Número de vezes executada no servidor

   c) [0,4] As mensagens de resposta do servidor são perdidas pela rede excedendo o *timeout* do cliente. O *timeout* do cliente é 1s e a repetição demora 100 ms.

Resultado para o cliente	Número de vezes executada no servidor

3) [0,6] Se a semântica fosse "no-máximo-uma-vez" (*at-most-once*) qual (ou quais) dos três quadros anteriores seria diferente? Identifique claramente qual ou quais e justifique.

Grupo II [3,5 valores]

Considere novamente **o programa em Java do Grupo I**. Mantendo o serviço descrito pretende-se agora que o sistema use RMI e as funcionalidades deste sistema de objetos distribuídos.

1) [0,6] Defina a interface `ICar` com os seguintes requisitos:
   - Os objetos da classe correspondente ficam residentes nos sistemas informáticos dos automóveis, mantendo-se ativos e enviando periodicamente a localização GPS do veículo.
   - O objeto tem, entre outros, um método:

```
CarReservation reserve(int userId) throws CarNotAvailable
```

   Este método deverá poder ser executado remotamente pelo cliente para reservar um carro. O parâmetro de resposta é uma reserva que permite reservar um carro durante 1 hora quando apresentado ao sistema do carro. Ignore, por agora, a informação que compõe uma reserva.

```
public interface ICar
```

2) Suponha que o `CarReservation` deve conter `int pinCode, DigSign pinDigSign`
   a) [0,4] Na lógica desta aplicação, este objeto deveria ser passado por valor ou por referência? Justifique.

   b) [0,5] Que diferença existe a nível da programação para especificar a decisão da alínea anterior?

3) O cliente do sistema obtém de acordo com a alínea anterior uma referência remota para um objeto, podemos supor neste caso o `car1234`.
   a) [0,5] Comente a afirmação: "Para obter esta referência remota o objeto car1234 tem de registar-se no *RMI Registry* de outra forma não será possível encontrar a sua referência remota".

   b) [0,4] **Quando** é criada a referência remota inicial para este objeto? Justifique.

   c) [0,4] **Quem** é responsável por criá-la? Justifique.

   d) Suponha que o sistema implementa um *garbage collector* baseado na contagem de referências. Na situação descrita nesta alínea procure explicar:
      i) [0,3] Que valor terá o contador de referências na máquina virtual onde se executa o objeto `car1234`?

      ii) [0,4] Justifique a sua resposta detalhando os passos que permitem calcular esse valor.

Grupo III [3,5 valores]

Considere um sistema de encomendas eletrónicas baseado na tecnologia de **Web Services**.

1) O Servidor recebe pedidos de encomenda através da operação `placeOrder`.
   Capturou-se na rede a seguinte mensagem SOAP a caminho do Servidor:

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns2:placeOrder xmlns="http://tempuri.org/PurchaseOrderSchema.xsd"
xmlns:ns2="http://ws.example/">
            <ns2:order OrderDate="2016-06-12+01:00">
                <ShipTo>
                    <name>Warehouse</name>
                    <street>22nd st</street>
                    <city>Springfield</city>
                    <state>MA</state>
                    <zip>1105</zip>
                </ShipTo>
                <BillTo>
                    <name>Headquarters</name>
                    <street>Vassar St</street>
                    <city>Cambridge</city>
                    <state>MA</state>
                    <zip>2139</zip>
                </BillTo>
            </ns2:order>
        </ns2:placeOrder>
    </S:Body>
</S:Envelope>
```

   a) [0,3] Consegue inferir a linguagem de programação em que o Servidor está programado? Justifique.

   b) [0,3] O que define o valor "`http://tempuri.org/PurchaseOrderSchema.xsd`" na mensagem?
      i) O endereço de destino da mensagem SOAP.
      ii) A localização do *schema*.
      iii) O espaço de nomes dos elementos `ShipTo` e `BillTo`.
      iv) O espaço de nomes do elemento `order`.

2) Considere agora que o Cliente e o Servidor foram desenvolvidos em **Java** com a biblioteca JAX-WS.
   a) Escreva em pseudo-código as classes Java geradas para transporte de dados:
      i) [0,4] Classe que representa a Encomenda (`order`).

      ii) [0,5] Classe que representa a Morada. Pode assumir que `BillTo` e `ShipTo` são do mesmo tipo.

   b) Entretanto obteve acesso ao contrato do serviço que descreve a operação `placeOrder` e encontrou a seguinte informação: `<fault message="tns:orderFault" name="orderFault" />`
      i) [0,4] Como representaria o método Java correspondente à operação `placeOrder` na interface gerada tendo em conta esta definição? Assuma que a operação não retorna resultados.

```
void placeOrder(
```

      ii) [0,3] Dê exemplo de uma situação concreta em que faça sentido o Servidor devolver uma `orderFault`?

3) Considere os seguintes nomes de etiquetas XML usadas por normas de Web Services.

a) [0,3] Assinale com ● os elementos abaixo que são usados num documento WSDL.

`handlerChain`	`portType`
`message`	`service`
`header`	`envelope`
`types`	`binding`

b) [0,3] Que elementos escolhidos definem a interface abstrata do serviço no WSDL?

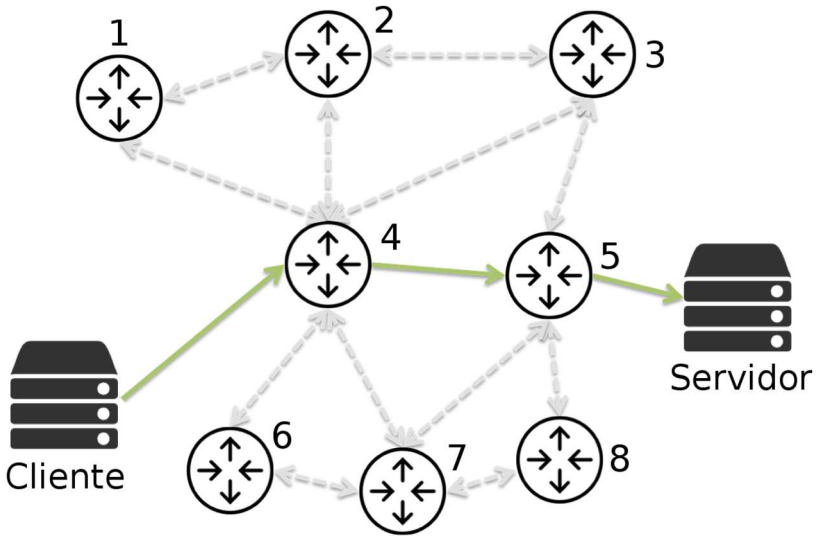
c) [0,4] Porque é necessária uma interface concreta no WSDL além da interface abstrata?

d) [0,3] Indique um item de informação que faça parte apenas da interface concreta do serviço no WSDL e explique sucintamente a utilidade dessa informação.

Grupo IV [4 valores]

Considere novamente o **sistema** de encomendas eletrónicas do **Grupo III**.

O Cliente e o Servidor estão interligados através da Internet, por nós de rede em que não confiam.
O fornecedor recebe encomendas através da operação `placeOrder` invocada por HTTP.



1) Um **atacante** capturou a mensagem SOAP apresentada em III 1) (pedido de `placeOrder`).

   a) [0,4] Descreva os passos necessários para realizar um ataque que leve a encomenda a ser entregue noutro endereço postal. Indique explicitamente onde se deve posicionar o atacante.

   b) [0,3] Se o pedido estivesse a ser enviado via HTTPS em vez de HTTP, continuaria a ser possível o mesmo ataque? Responda esclarecendo se a segurança oferecida pelo HTTPS é "ponto-a-ponto" ou "extremo-a-extremo" explicando a diferença.

2) Considere agora que tem à sua disposição **criptografia assimétrica**.

   a) [0,6] O que acrescentaria à mensagem para garantir *apenas* a **integridade**.

   Use { } para representar cifra e Kpub e Kpriv para indicar chave pública e privada.
   Acrescente uma legenda para outros elementos adicionais, caso sejam necessários.

```
<soap:envelope>
    <soap:header>









    </soap:header>
    <soap:body>
        <placeOrder .../>
    </soap:body>
</soap:envelope>
```

   b) [0,4] Qual seria a distribuição de chaves que permitiria garantir o **não-repúdio** dos pedidos de encomenda?

      i) A chave pública deve ser certificada por 2 níveis hierárquicos de CA.
      ii) A chave pública deve ser apenas do conhecimento do destinatário.
      iii) Deve usar-se um par de chaves diferente para cada mensagem.
      iv) A chave privada deve ser apenas do conhecimento do próprio.

c) Pretende-se agora usar **cifra híbrida** da seguinte forma:

{ M } K1, { K1 } K2 onde M representa a mensagem, { } cifra, e K1 e K2 são chaves criptográficas.

i) [0,3] Escolha um algoritmo de cifra para K1. Justifique.

ii) [0,3] Escolha um algoritmo de cifra para K2. Justifique.

3) Considere agora que está a utilizar o sistema **Kerberos V5**.

a) [0,4] Seguindo o protocolo Kerberos V5, um cliente C obteve um ***ticket*** para sessão com o serviço S. O cliente pretende agora enviar um pedido confidencial a S. Para tal, a mensagem deve incluir:

i) {pedido}Kpub s
ii) {pedido}Kcs
iii) {pedido}Kpriv s
iv) {pedido}Ksaut

b) [0,5] Usando a chave escolhida na resposta anterior, poderia garantir a **integridade** da mensagem?
Se sim, indique o nome do mecanismo e explique-o sucintamente.
Se não, indique a razão que impede a garantia.

c) [0,4] Considere o **autenticador** usado no Kerberos V5: $auth_{x,y} = \{x, T_{req}\}_{Kx,y}$

i) x é o resumo do nome do cliente, Treq é um valor numérico, Kx,y é uma chave simétrica.
ii) x é um nonce, Treq é marca temporal para garantir frescura, Kx,y é a chave pública de Y.
iii) x é a chave do cliente, Treq é marca temporal para garantir frescura, Kx,y é a chave de sessão.
iv) x é o nome do cliente, Treq é marca temporal para garantir frescura, Kx,y é a chave de sessão.

d) [0,4] Que implicação tem a utilização do valor **Treq** sobre os clientes e servidores envolvidos?

Grupo V [3 valores]

1) Considere um sistema cliente-servidor com **replicação ativa** com *quorum consensus*.
Onde: C1 – cliente com id 1; C2 – cliente com id 2; FE1 – *Front-end* de C1; FE2 – Front-end de C2;
R1, R2, R3 – réplicas do servidor; x uma é variável inteira.
Registaram-se as seguintes mensagens:
   - C1 -> FE1 read(x)
   - FE1 -> R1 read(x)
   - FE1 -> R2 read(x)
   - FE1 -> R3 read(x)
   - R2->FE1 x=0, <1,33>
   - R3->FE1 x=-100, <2,31>

   a) [0,4] O que são os valores <… , …> (exemplificados acima por <1,33> e <2,31>)? E para que servem?

   b) [0,3] Considera que a leitura de C1 poderia ser finalizada no passo seguinte a R3->FE1? Justifique.

   c) [0,6] C1 pretende agora executar ***write(x, 10)*** e C2 **write(x, 20)**.
   Considere que:
   - A leitura anterior já terminou e nenhuma outra operação se executou entretanto.
   - R2 teve uma falta silenciosa (R2 KO) e antes não tinham ocorrido outras faltas.
   - A ligação de rede do FE1 a todas as réplicas tem aproximadamente o dobro da latência da ligação de rede de FE2 a todas as réplicas.

   Construa uma possível sequência de mensagens até à conclusão das duas escritas, caso seja possível.
   Inclua os meta-dados <… , …> relevantes na sua resposta e pode acrescentar mais • se necessário.
      - **R2 KO**
      - C1 -> FE1 write(x, 10)
      - C2 -> FE2 write(x, 20)
      -
      -
      -
      -
      -
      -
      -

2) Considere agora um outro sistema replicado, baseado no protocolo ***primary-backup*** com um servidor de nomes. Onde: C – cliente; FE – front-end; S – serviço; R1 – réplica primária; R2 – réplica secundária; NS – servidor nomes. a é uma variável inteira.
O *front-end* resolve *sempre* o nome do servidor antes de o contactar através da função *resolve()* que devolve o endereço de uma réplica – *addressOf()*.

Registaram-se as seguintes mensagens:

- C -> FE read(a)
- R1 -> R2 …
- FE -> NS resolve(S)
- NS -> FE addressOf(R1)
- FE -> R1 read(a)
- R1 -> FE a=100
- R1 -> R2 …
- FE -> C a=100

a) [0,3] Qual é o conteúdo e propósito das mensagens R1 -> R2 … ?

b) [0,4] Antes da resposta ao cliente, não deveria ter surgido R1 -> R2 read(x)?
Considera este comportamento normal ou uma falta? Justifique.

c) [0,6] O cliente pretende agora executar ***write(a, 300)***.
Complete uma possível sequência de mensagens sendo que R1 KO representa a falta silenciosa de R1.
Complete a sequência até que o sistema recupere da falta (caso recupere).

- C -> FE write(a, 300)
- FE -> NS resolve(S)
- NS -> FE addressOf(R1)
- FE -> R1 write(a, 300)
- **R1 KO**
-
-
-
-
-
-
-
-

d) [0,4] Qual é o cálculo correto para o tempo de recuperação do serviço assumindo *primary-back com naming server*?

i) (P + tmax) + (tpublishNS + tresolveNS) + tretryFE

ii) P + tpublishNS + tretryFE

iii) (P + tmax) + tpublishNS + tretryFE + tresponse

iv) (P + 3*tmax) + (tpublishNS + tresolveNS) + tretryFE

Grupo VI [1,5 valores]

Numa plataforma de *crowdfunding* cada projeto proposto recebe promessas de donativos monetários vindas de diferentes interessados em apoiar o projeto. Após um projeto angariar o montante pretendido, é realizada a transferência de cada donativo para a conta do proponente do projeto.

Assuma que para um dado projeto as promessas de donativos foram recolhidas na lista donativos, diferentes donativos podem vir de diferentes bancos (*d.bank*). Cada banco tem um serviço remoto que oferece a operação *transfer* e mantém as suas contas num sistema de dados transacional.

```
1     boolean collectDonations(List<Donation> donations, Account proponentAccount) {
2           Object tx = openTransaction();
3           for each (Donation d in donations) {
4                 d.bank.transfer(d.account, d.amount, proponentAccount, tx);
5                 if (error) {
6                       return error;
7                 }
8           }
9           return closeTransaction(tx);
10    }
```

1) [0,4] O programa acima poderá violar a propriedade da Atomicidade.
Indique como corrigiria o programa para garantir a propriedade da Atomicidade.
Programe as alterações indicando o número ou números das linhas a modificar ou a corrigir.

2) O valor retornado na linha 2 (tx)

a) [0,2] Quem atribui o valor a tx?

b) [0,3] Que utilidade tem tx na sequência da transação, no caso concreto da linha 4? Justifique.

3) Assuma que o programa é executado com 4 bancos participantes (B1, B2, B3, B4) e chega à linha 9 do programa. Indique qual a decisão tomada pelo coordenador em cada uma das seguintes situações:
   a) [0,3] Após enviar canCommit(tx), o coordenador recebeu voto Yes de todos os participantes exceto do B1 que não respondeu tendo expirado o *timeout*.
      i) O Coordenador envia doAbort a todos.
      ii) O Coordenador envia doCommit a todos.
      iii) O Coordenador envia doCommit aos participantes que votaram Yes e doAbort a B1.
      iv) O Coordenador não toma nenhuma decisão neste caso ficando bloqueado.

   b) [0,3] Após enviar canCommit(tx), o coordenador recebeu voto Yes de todos os participantes e agiu em conformidade, B1 não volta a contactar o coordenador.
      i) O Coordenador envia doAbort a todos.
      ii) O Coordenador envia doAbort a B1.
      iii) O Coordenador mantém a transação aberta esperando que B1 recupere.
      iv) O Coordenador envia uma excepção ao cliente e fecha a transação.

Grupo VII [1,5 valores]

1) [0,3] Considere o seguinte binding em Java RMI e o nome "`//xpto.jogox`"

```
Jogo j = (DeckofCards) Naming.lookup("//xpto.jogox");
```

   a) É um URL que localiza diretamente o objeto servidor
   b) O nome identifica o objeto no RMI registry local e permite obter a sua referência remota
   c) O nome é local à máquina virtual onde está a correr o servidor
   d) O nome é puro e pode estar localizado em qualquer RMI registry

O seguinte excerto faz parte da página de informação do Fénix. A informação apresentada deve ser percebida tendo em conta o que aprendeu na cadeira de Sistemas Distribuídos.

> **Sistema de Autenticação Centralizada**
>
> Inclui o diretório central do IST através do protocolo LDAP;
>
> O sistema de autenticação Kerberos;

2) Um serviço de diretório é diferente de um serviço de nomes
   a) [0,3] Indique claramente a principal diferença entre um serviço de diretório e um serviço de nomes.

b) [0,3] A que se refere o protocolo LDAP da página do IST? Explique claramente o que deverá conter este diretório. Procure ilustrar com a sua informação como utente do IST.

c) [0,3] No seu projeto usou também um serviço de diretório.
Qual a principal vantagem que obteve da sua utilização. Justifique.

3) [0,3] LDAP e certificados digitais
   a) O diretório pode ter os certificados digitais X509 dos utilizadores.
   b) Não pode colocar certificados no diretório porque existe a possibilidade de ataque de "*man-in-the-middle*".
   c) O LDAP não resulta da evolução do X500 pelo que não consegue armazenar certificados X509 que apenas podem existir nesse diretório.
   d) O LDAP tem os certificados porque é uma *Certification Authority* (neste caso do IST).