Número: Nome:

# LEIC/LERC – 2009/10

## Primeiro Teste de Sistemas Distribuídos

23 de Abril de 2010

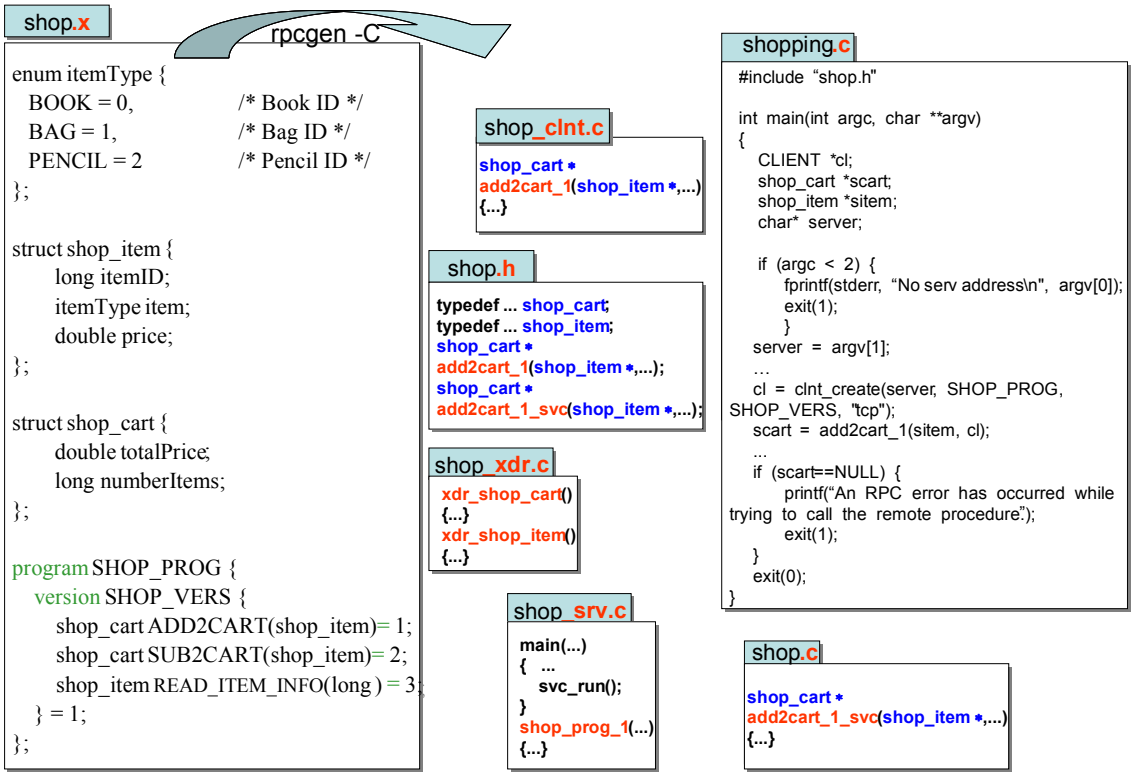
Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas.

Duração: 1h30m

---

## Grupo I [6 v]

1) Considere o seguinte código fonte de uma aplicação cliente-servidor programado em Sun RPC.




a. [0,6 v] Descreva sucintamente o que faz este programa distribuído.

b. [1,1 v] Indique, em relação à Figura, qual/quais dos ficheiros apresentados:

i. Incluem código introduzido manualmente pelo programador.

ii. São necessários para compilar a aplicação cliente.

iii. São necessários para compilar a aplicação do servidor

iv. Onde é efectuado o Binding do cliente ao servidor (ficheiro(s) e instrução(ões)).

v. Onde é efectuada a chamada do procedimento remoto no programa cliente (ficheiro e instrução).

2) Considere o stub do cliente detalhado ("client stub"):

**shop_clnt.c**

```
#include "shop.h"

static struct timeval TIMEOUT = { 25, 0 };

shop_cart *
add2cart_1(shop_item *argp, CLIENT *clnt)
{
          static shop_cart clnt_res;
          if (clnt_call(clnt, ADD2CART,
                    xdr_shop_item, argp,
                    xdr_shop_result, &clnt_res,
                    TIMEOUT) != RPC_SUCCESS) {
        return (NULL);
          }
          return (&clnt_res);
}

shop_cart *
sub2cart_1(shop_item *argp, CLIENT *clnt) {...}
shop_item *
read_item_info_1(long *argp, CLIENT *clnt) {...}
```

a. [0,9 v] Como é que o stub do cliente ("client stub") sabe onde se executa o servidor? Esta solução privilegia a transparência? Justifique.

b. [0,7 v] Qual das afirmações é verdadeira? (Resposta errada desconta ¼ da pergunta.)
**xdr_shop_item** é um ponteiro para uma rotina de marshalling baseada em XDR, a qual utiliza:

i. Estrutura das mensagens implícita e política de conversão dos dados para formato canónico
ii. Estrutura das mensagens implícita e política de conversão dos dados o-receptor-converte
iii. Estrutura das mensagens explícita e política de conversão dos dados para formato canónico
iv. Estrutura das mensagens explícita e política de conversão dos dados o-receptor-converte

Número: Nome:

3) Relativamente à execução distribuída de uma chamada remota:
   a. [0,9 v] Diga qual a semântica de execução oferecida pelos seguintes RPCs:
      i. Cliente envia cada pedido apenas uma vez para o socket TCP e bloqueia-se à espera da resposta. Se o socket for fechado antes da resposta chegar, o RPC retorna erro à aplicação. Assuma que todas as falhas de comunicação são toleradas pelo socket TCP.

| |
|---|
| |

      ii. Cliente envia cada pedido para o socket UDP e bloqueia-se à espera da resposta. Caso não receba a resposta ao fim de um determinado tempo, o cliente reenvia a mensagem a receber.

| |
|---|
| |

   b. [0,9 v] Considere um cliente que pretende invocar o procedimento remoto ADD2CART. Indique qual das opções acima (a.i ou a.ii) seria mais apropriada. Justifique sucintamente.

| |
|---|
| |
| |
| |

   c. [0,9 v] Considere agora a invocação do procedimento remoto READ_ITEM_INFO. Indique qual das opções acima (a.i ou a.ii) seria mais apropriada. Justifique sucintamente.

| |
|---|
| |
| |
| |

# Grupo II [4 v]

Considere o seguinte WSDL:

```
<?xml version="1.0" encoding="UTF-8"?> <definitions name="Shop"
        targetNamespace="http://exemploTesteSD.com/shop/shop"
        xmlns:tns="http://exemploTesteSD.com/shop/shop"
        xmlns="http://schemas.xmlsoap.org/wsdl/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">

        <types> <xsd:schema elementFormDefault="qualified" targetNamespace="http://exemploTesteSD.com/shop/shop">
                <xsd:simpleType name="enumType"> <xsd:restriction base='xsd:string' >
                        <xsd:enumeration value="BOOK" />
                        <xsd:enumeration value="BAG" />
                        <xsd:enumeration value="PENCIL" />
                </xsd:restriction> </xsd:simpleType>
                <xsd:complexType name="ShopItemType"> <xsd:sequence>
                        <xsd:element name="itemID" type="xsd:long"/>
                        <xsd:element name="itemType" type="tns:enumType"/>
                        <xsd:element name="price" type="xsd:double"/>
                </xsd:sequence> </xsd:complexType>
                <xsd:complexType name="ShopItemListType"> <xsd:sequence>
                        <xsd:element name="ShopItem" minOccurs="0" maxOccurs="unbounded" type="tns:ShopItemType"/>
                </xsd:sequence> </xsd:complexType>
                <xsd:complexType name="ShopCartType"> <xsd:sequence>
                        <xsd:element name="totalPrice" type="xsd:double"/>
                        <xsd:element name="numberItems" type="xsd:long"/>
                </xsd:sequence> </xsd:complexType>
        </xsd:schema> </types>

        <message  name="ShopItemMessage"> <part name="ShopItem" type="tns:ShopItemType"/>  </message>
        <message  name="AllShopItemMessage"> <part name="ShopItemList" type="tns:ShopItemListType"/>  </message>
        <message  name="ShopCartMessage"> <part name="ShopCart" type="tns:ShopCartType"/>  </message>
        <message  name="None">  </message>
        <message  name="ItemMessage">  <part name="Item" type="xsd:long"/>  </message>

        <portType name="ShopPortType">
                <operation name="add2CART">
                        <input message="tns:ShopItemMessage"/>
                        <output message="tns:ShopCartMessage"/>
                </operation>
                <operation name="sub2CART">
                        <input message="tns:ShopItemMessage"/>
                        <output message="tns:ShopCartMessage"/>
                </operation>
                <operation name="read_ITEM_INFO">
                        <input message="tns:ItemMessage"/>
                        <output message="tns:ShopItemMessage"/>
                </operation>
                <operation name="getALLITEMS">
                        <input message="tns:None"/>
                        <output message="tns:AllShopItemMessage"/>
                </operation>
        </portType>

        <binding name="ShopBinding" type="tns:ShopPortType">
                <soap:binding style ="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
                <operation name="add2CART">
                        <soap:operation soapAction=""/>
                        <input>  <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </input>
                        <output> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </output>
                </operation>
                <operation name="sub2CART">
                        <soap:operation soapAction=""/>
                        <input> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </input>
                        <output> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </output>
                </operation>
                <operation name="read_ITEM_INFO">
                        <soap:operation soapAction=""/>
                        <input> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </input>
                        <output> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </output>
                </operation>
                <operation name="getALLITEMS">
                        <soap:operation soapAction=""/>
                        <input> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </input>
                        <output> <soap:body use="literal" namespace="http://exemploTesteSD.com/shop/shop"/> </output>
                </operation>
        </binding>

        <service name="ShopService">
                <documentation>My test one service</documentation>
                <port name="ShopPort" binding="tns:ShopBinding">
                        <soap:address location="http://exemploTesteSD.com/shop/endpoint"/>
                </port>
        </service>
</definitions>
```

Número: Nome:

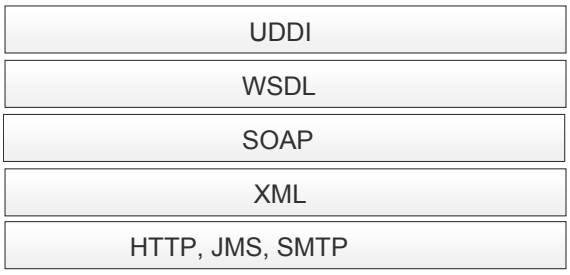
1) Os Web Services podem ser vistos como uma plataforma de RPC, na mesma categoria de plataformas mais antigas como o SUN RPC ou o DCE RPC. A figura indica a descrição do serviço do Grupo I, na IDL dos WebServices: o WSDL, e adicionada da operação getALLITEMS.

   a. [0,6 v] Indique a assinatura das função getALLITEMS do serviço definido no documento.

   b. [0,6 v] Indique qual a parte abstracta e a parte concreta da interface WSDL.

   c. [1,1 v] Considere que a operação READ_ITEM_INFO deste serviço deveria receber um parâmetro de entrada adicional (o do itemType) para além do identificador do item (itemID). Considere também que, em alguns casos, a operação pode falhar devido a falha na base de dados do servidor. Indique quais secções deve modificar, reescrevendo a parte estritamente necessária.

2) [0,6 v] Considere as camadas da pilha de protocolos dos WebServices da Figura.




   Indique de forma sucinta e objectiva os serviços disponibilizados por cada uma destas camadas.

```
POST /shop HTTP/1.1
Host: www. exemploTesteSD.com
Content-Type: text/xml; charset="utf-8"
Content-Length: 322
SOAPAction: "http://exemploTesteSD.com/shop/SENDSHOP"
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema"
         xmlns:ns1="http://exemploTesteSD.com/shop">
    <soapenv:Body>
        <ns1:SENDSHOP>
            <ns1:envelope> Is SD book available for buying? </ns1: envelope>
        </ns1: SENDSHOP>
    </soapenv:Body>
```

3) [1,1 v] Considere o documento indicado acima. Suponha que se pretende:
    1. incluir um timestamp nas mensagens SOAP
    2. fazer um log de todas as mensagens recebidas pelo servidor
    3. manter uma estatística sobre a distribuição da proveniência geográfica dos pedidos de compras (como o pedido apresentado acima).

    Idealmente, tal deveria ser conseguido sem alterar a interface nem a implementação do serviço já existente. Indique de forma sucinta como asseguraria este objectivo, referindo de forma clara quais as alterações que introduziria (i) no cliente, (ii) na mensagem SOAP e (iii) no servidor.

    a. No cliente:

    b. Na mensagem SOAP:

    c. No servidor:

Número: Nome:

# Grupo III [5 v]

Considere as seguintes declarações de uma classe abstracta em Java:

```
1 public interface Shape extends Remote {
2    int getVersion() throws RemoteException;
3    GraphicalObject getAllState() throws RemoteException;
4 }

6 public interface ShapeList extends Remote {
7       Shape newShape(GraphicalObject g) throws RemoteException;
8       Vector allShapes()throws RemoteException;
9       int getVersion() throws RemoteException;
10 }
```

E a seguinte classe que descreve o código do main do servidor:

```
11 public class ShapeListServer {
12    public static void main(String args[]){
13       System.setSecurityManager(new RMISecurityManager());
14       try{
15             ShapeList aShapelist = new ShapeListServant();
16             Naming.rebind("ShapeList", aShapelist);
17     } catch(Exception e) {
18             System.out.println("ShapeList server main " +
19                                e.getMessage());}
20     }
21 }
```

1) Considere a classe que descreve o `main` do servidor
   a. [0,6 v] Explique o que sucede na linha 15 e qual a sua relação com os termos habituais de descrição destes sistemas: “servidor” e “interface do servidor”.

| |
|---|
| |
| |
| |

   b. [0,6 v] Como é que os clientes obtêm no sistema distribuído a referência remota para o objecto servidor?
   Explique que linhas de código justificam a sua resposta e que entidades ai estão envolvidas.

| |
|---|
| |
| |
| |
| |

2) Ambas as interfaces herdam de remote.
   a. [0,6 v] Que propriedades estão associadas a esta herança?

| |
|---|
| |
| |
| |

   b. [0,4 v] Por que razão é declarada uma excepção na definição destas interfaces? Justifique.

| |
|---|
| |
| |

3) Considere a interface do objecto servidor e a função `allShapes`.
   a. Suponha que já foram criadas três instâncias de Shape no servidor e adicionadas à ShapeList do servidor, e que o cliente invoca `allShapes`.
      i. [0,4 v] Que objectos deverão ser criados no lado do cliente? Justifique.

      ii. [0,4 v] Quantos? Justifique.

   b. Considerando que o protocolo de garbage collection é do tipo de contagem de referências.
      i. [0,4 v] O que terá de fazer o cliente na sequência da invocação anterior?

      ii. [0,4 v] Quando será eliminado cada objecto criado no cliente e referido na alínea a)?

      iii. [0,4 v] Quando esse objectos forem eliminados o que terá de fazer o cliente em relação ao protocolo de garbage collection?

4) Considere a função `newShape`, que tem como parâmetro de entrada um objecto `GraphicalObject` e cuja interface herda de serializable e não herda de remote.
   a. [0,4 v] Como é passado esse parâmetro do cliente para o servidor? Justifique.

   b. [0,4 v] O retorno dessa função é por sua vez um objecto `Shape`. Compare a passagem de parâmetro com a referida na alínea a anterior. É a mesma? Justifique.

Número: Nome:

# Grupo IV [5 v]

Considere o seguinte fragmento de um documento XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema"
         xmlns:ns1="http://xpto">

    <soapenv:Body>
        <ns1:QualBanco>
            <ns1:banco>BES</ns1:banco>
        </ns1:QualBanco >
    </soapenv:Body>
</soapenv:Envelope>
```

1) O nome "banco" é um nome do espaço de nomes XML que designa um elemento.
   a. [0,6 v] É um nome local ou global? Apresente um exemplo que justifique a sua resposta.

| |
|---|
| |
| |
| |

   b. [0,4 v] Que diferença existe em relação à resposta anterior para o nome "ns1:banco".

| |
|---|
| |
| |
| |

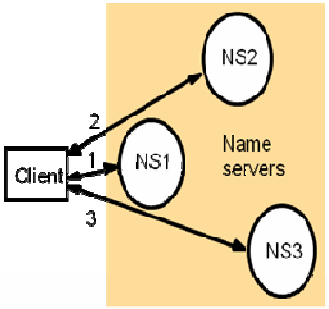
2) O qualificador ns1 corresponde a "http://xpto"
   a. [0,6 v] Qual a função deste URI, localizar ou identificar? Justifique.

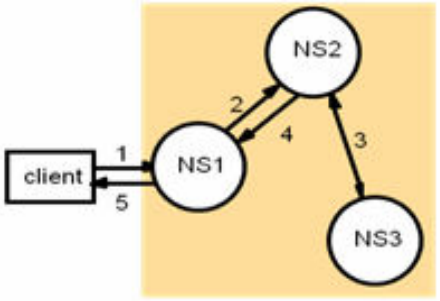
| |
|---|
| |
| |
| |

   b. [0,4 v] Como o deveria qualificar: URN ou URL?

| |
|---|
| |

3) Considere a seguinte figura que descreve duas formas de realizar a resolução de nomes.



a. [0,6 v] Estes esquemas estão ligados a uma forma de construção de nomes hierárquica ou são completamente independentes da estrutura dos nomes? Justifique.

b. O desempenho de um serviço de nomes pode ser significativamente melhorado uma vez que os nomes mudam com relativa pouca frequência

   i. [0,6 v] De que forma se pode, com base nessa propriedade, aumentar o desempenho dos pedidos de resolução feitos pelo cliente? Seja claro na sua resposta.

   ii. [0,6 v] Qual das duas formas de resolução apresentadas na figura acima acha que é mais eficaz para tirar partido do mecanismo referido na alínea anterior? Justifique.

c. O DNS é um serviço de nomes.

   i. [0,4 v] Como classifica os nomes DNS em termos de hierarquia, homogeneidade, pureza?

   ii. [0,4 v] Aplicam-se-lhe ou não os esquemas de resolução da figura?

   iii. [0,4 v] Explique se o diagrama procura ilustrar o mecanismo de servidor primário e secundário existente no DNS