



1)[3 val] Considere que tem de implementar objectos distribuídos que poderão ser alojados em sistemas existentes em localizações geograficamente dispersas. Indique, justificando, que cuidados deveria ter no desenho das interfaces desses objectos.

2)[3 val] Explique porque é que a comunicação por grupos com multicast fiável facilita a implementação de sistemas distribuídos com tolerância a falhas no acesso a serviços.

3)[2 val] Uma das características fundamentais dos sistemas distribuídos deverá ser a heterogeneidade. Apresente as razões pelas quais a utilização de Web Services na implementação de sistemas distribuídos facilita o suporte à heterogeneidade.

4) Considere um sistema distribuído em .NET Remoting de suporte a leilões. Existe um servidor que implementa a interface **IServer**.

Quando forem registados novos artigos (método **newArtigo**) para leiloar no servidor, este deverá invocar o método **leilao** nos clientes que tenham registado (método **interesse**) o seu interesse em participar nos leilões.

Para além de registarem o seu interesse em leilões, os clientes podem em qualquer altura licitar valores para artigos de que tenham sido notificados (método **leilao**) previamente.

```
public interface ILeilao
{
    void leilao(string artigo,
               decimal precoBase);
}

public interface IServer
{
    void interesse(ILeilao lei);
    void newArtigo(string artigo,
                  decimal precoBase);
    void licitar(string artigo,
                 decimal valor);
}
```

a) [2 val] Proponha o esboço em C# da classe que implementa o objecto servidor, justificando se optou pela opção de objectos *SAO SingleCall* ou *SAO Singleton*.

b) [2 val] Apresente o ficheiro de configuração do servidor, admitindo que os clientes e o servidor se encontram na mesma rede local.

c) [2 val] Apresente o troço de código C# de um cliente que inicia o acesso ao servidor e invoca a operação **interesse**.

5) Tendo como referência o ficheiro WSDL (Web Service Definition Language) de um *Web Service* .NET, fornecido em anexo, responda às seguintes questões:

a)[1.5 val] Interprete o elemento **binding** do ficheiro WSDL (dentro do rectângulo) indicando que informações se podem extrair desse elemento e que são importantes na caracterização das interações dos clientes com o *Web Service*.

a)[3 val] Apresente o esboço em C# de implementação em .NET do *Web Service* descrito no ficheiro WSDL.

c)[1.5 val] Apresente o troço de código de uma aplicação cliente que acede ao *Web Service* para invocar as operações que ele disponibiliza.

```

<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap=http://schemas.xmlsoap.org/wsdl/soap/ .....
                xmlns:tns=http://EXAME.SD0809/ targetNamespace=http://EXAME.SD0809/ >
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://EXAME.SD0809/">
    <s:element name="SendMsg">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="msg" type="tns:Msg" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:complexType name="Msg">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="numMsg" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="txt" type="s:string" />
      </s:sequence>
    </s:complexType>
    <s:element name="SendMsgResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SendMsgResult" type="s:boolean" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="Hdin" type="tns:Hdin" />
    <s:complexType name="Hdin">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="emissor" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="codes" type="tns:ArrayOfInt" />
      </s:sequence>
      <s:anyAttribute />
    </s:complexType>
    <s:complexType name="ArrayOfInt">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="unbounded" name="int" type="s:int" />
      </s:sequence>
    </s:complexType>
    <s:element name="Hdout" type="tns:Hdout" />
    <s:complexType name="Hdout">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="numMsg" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="myCodes" type="tns:ArrayOfInt" />
      </s:sequence>
      <s:anyAttribute />
    </s:complexType>
  </s:schema>
</wsdl:types>

```

(Continua)



(continuação)

```
<wsdl:message name="SendMsgSoapIn">
  <wsdl:part name="parameters" element="tns:SendMsg" />
</wsdl:message>
<wsdl:message name="SendMsgSoapOut">
  <wsdl:part name="parameters" element="tns:SendMsgResponse" />
</wsdl:message>
<wsdl:message name="SendMsgHdin">
  <wsdl:part name="Hdin" element="tns:Hdin" />
</wsdl:message>
<wsdl:message name="SendMsgHdout">
  <wsdl:part name="Hdout" element="tns:Hdout" />
</wsdl:message>

<wsdl:portType name="ServiceSoap">
  <wsdl:operation name="SendMsg">
    <wsdl:input message="tns:SendMsgSoapIn" />
    <wsdl:output message="tns:SendMsgSoapOut" />
  </wsdl:operation>
```

```
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="SendMsg">
    <soap:operation soapAction="http://EXAME.SD0809/SendMsg" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
      <soap:header message="tns:SendMsgHdin" part="Hdin" use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
      <soap:header message="tns:SendMsgHdout" part="Hdout" use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

```
<wsdl:service name="Service">
  <wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
    <soap:address location="http://localhost:49468/WebSite1/Service.asmx" />
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>
```