

GENÉRICAS

[2 val] A transparência é uma das características fundamentais de um sistema distribuído. Comente as características da plataforma .NET Remoting relativamente aos seguintes tipos de transparência: **Acesso**, **Localização** e **Migração** de objectos.

[2 val] Refira os principais aspectos que diferenciam os objectos distribuídos relativamente ao paradigma de programação orientado aos objectos.

[2 val] Considere uma aplicação cliente/servidor que foi desenvolvida em .NET Remoting, usando objectos *Server Activated Objects* (SAO) em modo *Singleton*. Com o aumento do número de utilizadores a aplicação começa a revelar problemas de desempenho pelo que surge a necessidade de expandir a componente servidora. Comente e justifique se utilizaria uma estratégia de expansibilidade Vertical ou expansibilidade Horizontal.

[2 val] Considere um programador que desenvolveu uma aplicação cliente/servidor usando objectos distribuídos **stateful**. Ao fim de alguns meses de utilização, constatou-se que era necessário expandir o sistema para melhorar o desempenho. Indique, justificando, se deve ser seguida uma estratégia de expansão vertical ou horizontal.

COMUNICAÇÃO POR GRUPOS

[2 val] Considere o desenvolvimento de um serviço de nomes que só fica indisponível se falharem as N réplicas ($N \geq 3$). Considere que cada uma das réplicas do sistema é executado em máquinas independentes ligados em rede (com redundância).

Discuta a vantagem da utilização de uma plataforma que suporta grupos de processos onde é disponibilizada comunicação com ordenação total, admitindo que apenas uma das réplicas é responsável pela resposta aos pedidos.

[2 val] Apresente as vantagens em utilizar comunicação por grupos e algoritmos de eleição na implementação de serviços tolerantes a falhas.

TEMPO E COORDENAÇÃO

[3 val] Considere um sistema distribuído, constituído por 4 processos, que utiliza comunicação por *multicast* e um mecanismo baseado em relógios vectoriais que garante ordenação causal de eventos. O estado dos relógios locais é: **p0=[2,4,4,3]** ; **p1=[2,5,3,3]** ; **p2=[1,4,4,3]** ; **p3=[2,4,4,5]**.

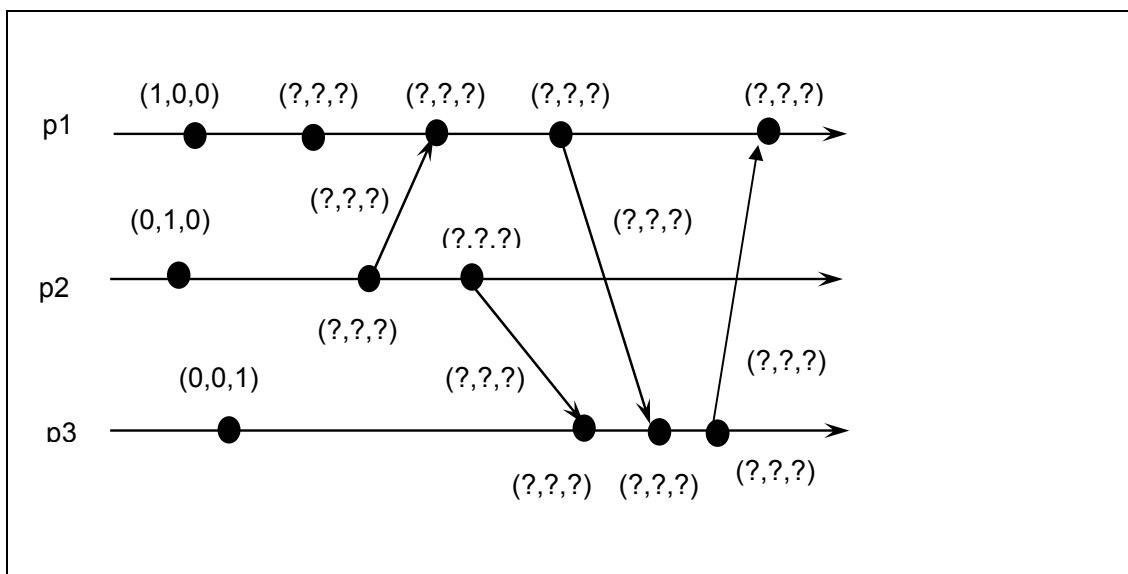
Considerando que **p0** vai enviar uma mensagem *multicast* para **p1,p2,p3**, apresente qual o vector de relógios que é enviado na mensagem justificando o que acontece na recepção da mensagem em **p1,p2 e p3**.

SISTEMAS DISTRIBUÍDOS

Exemplos de Questões para Exame

[3 val] O algoritmo de exclusão mútua de *Ricart e Agrawala* é totalmente distribuído. No entanto, a sua implementação requer a satisfação de requisitos com características específicas. Apresente justificando quais são esses requisitos.

[3 val] Assumindo que os três pares usam o modelo de relógios lógicos vectoriais e realizam a comunicação apresentada na figura, complete justificando os vectores apresentados por $(?, ?, ?)$.



[2 val] Considere um sistema distribuído onde é usado o algoritmo Ricart&Agrawala para acesso em exclusão mútua a um recurso crítico. No entanto, verifica-se que o sistema é ineficiente pois tipicamente um processo acede várias vezes ao recurso antes de qualquer outro processo o necessitar. Justifique porque é que o algoritmo Ricart&Agrawala provoca a ineficiência e apresente uma possível alteração ao algoritmo, para que passe a contemplar o padrão do sistema atrás descrito e portanto seja mais eficiente.

[2 val] Considere um sistema distribuído que utiliza um algoritmo de exclusão mútua centralizado. Comente o que pode acontecer e formas de o evitar se o processo que detém o Token (acesso ao recurso partilhado) falhar.

[2 val] Relativamente à opção pelo algoritmo de exclusão mútua Centralizado ou o algoritmo em Anel, discuta as vantagens/desvantagens de cada um nos seguintes dois cenários: **a)** Os participantes acedem à região crítica de forma obsessiva; **b)** Os participantes raramente acedem à região crítica.

.NET REMOTING

[2 val] Em ambientes cliente/servidor, existe, por vezes, a necessidade do servidor fazer também chamadas ao cliente (mecanismo de *callback*). Apresente quais os mecanismos que podem ser usados em .NET *Remoting* para implementar *callbacks*.

[2 val] Num ambiente de objectos distribuídos existe a dificuldade de gerir o tempo de vida dos objectos e consequentemente realizar de forma eficiente o *garbage collection*. Apresente o modelo existente em .NET *Remoting* para ultrapassar essa dificuldade.

Pretende-se implementar em ambiente distribuído utilizando-se .NET *Remoting*, uma solução cliente/servidor para gerir filas de espera numa organização e com as seguintes especificações:

- O servidor suporta operações para dois tipos de aplicações cliente:

Cliente de registo de pessoas:

Registrar – Dado o N° do Bilhete de Identidade e um de 4 possíveis assuntos, devolve o N° de ordem de atendimento e o N° do guichet;

Desistir – Dado o N° de ordem e o N° de guichet permite que um utilizador desista;

Cliente de chamada para atendimento:

Atender – Dado o N° de guichet activa num display a chamada do próximo cliente a atender;

- [2 val]** Especifique a interface em linguagem C# necessária à solução pretendida, colocando em comentário a descrição e pressupostos que considerou.
- [2 val]** Discuta a implementação do servidor como um objecto *SAO SingleCall* ou *SAO Singleton*.
- [2 val]** Apresente o esqueleto do servidor, os ficheiros de configuração do servidor e do cliente, bem como o troço de código do cliente que inicia o acesso ao serviço e invoca a operação Registrar.

Pretende-se implementar em ambiente distribuído utilizando-se .NET *Remoting*, uma solução cliente/servidor para gerir o acesso a contas bancárias e com as seguintes especificações:

- O servidor suporta operações para dois tipos de aplicações cliente:

Cliente de Consultas:

Saldo – Dado o NIB da conta devolve o saldo existente na mesma;

Movimentos – Dado o NIB da conta permite obter os últimos movimentos (máximo 10) existentes nessa conta. Um movimento é caracterizado por: data, descrição e valor;

Cliente para efectuar transacções:

PagarServiço – Dado o NIB da conta permite efectuar o pagamento de um serviço. Um serviço caracteriza-se por: Código entidade, Referência e Valor;

SISTEMAS DISTRIBUÍDOS

Exemplos de Questões para Exame

Transferência – Dados dois NIB de duas contas, permite efectuar uma transferência interbancária entre duas contas de um determinado valor;

- a) [2 val] Especifique a interface em linguagem C# necessária à solução pretendida, colocando em comentário a descrição e pressupostos que considerou.
- b) [2 val] Apresente vantagens/desvantagens de implementar o servidor como um objecto *SAO SingleCall* ou *SAO Singleton*, admitindo que esse objecto acede a uma base de dados de contas bancárias e que poderão existir um número muito elevado de clientes concorrentes.
- c) [2 val] Apresente o esqueleto do servidor, os ficheiros de configuração do servidor e do cliente, bem como o troço de código do cliente que inicia o acesso ao serviço e invoca a operação **PagarServiço**.

[2 val] Considere um servidor de gestão de alunos implementado em *.NET Remoting* que permite alterar os dados do aluno (nome, morada, BI, N° Contribuinte ...etc). Considere o seguinte código do cliente para iniciar a alteração dos dados de um aluno:

DadosAluno dal = serverObj.GetDadosAluno(numeroAluno)

Comente, justificando, se o tipo **DadosAluno** deverá ser um objecto *Marshal By Reference* ou *Marshal By Value*, exemplificando em C# a continuação do código para permitir alterar os dados do aluno.

Considere que na máquina com nome **Banco**, existe uma aplicação servidora com o ficheiro de configuração abaixo apresentado e que disponibiliza um objecto remoto que implementa a interface **IContaBank**.

<pre>namespace Exame { public interface IContaBank { int saldo {get;}; void depositar(int valor); bool levantar(int valor); } } // disponível no assembly // IContaBank.dll</pre>	<pre><configuration> <system.runtime.remoting> <application> <channels> <channel ref="http" port="2004" /> </channels> <service> <wellknown type="Exame.ContaBank, Server" mode="Singleton" objectUri="RemContaBank" /> </service> </application> </system.runtime.remoting> </configuration></pre>
---	---

- a)[2 val] Apresente o ficheiro de configuração de uma aplicação cliente que pretende aceder ao servidor.
- b)[2 val] Implemente a aplicação cliente que permite **levantar** 100 Euros se o **saldo** da conta for superior a 500 Euros.

SISTEMAS DISTRIBUÍDOS

Exemplos de Questões para Exame

c)[2 val] Discuta as alterações que teria de efectuar, incluindo o servidor, e as implicações daí decorrentes se o objecto remoto fosse **SingleCall** e o protocolo utilizado fosse **tcp**.

d)[2 val] O tempo de vida de um objecto remoto é, por omissão, limitado, podendo no entanto ser controlado através de um objecto *Sponsor* residente no lado do cliente ou no lado do servidor. Proponha e justifique uma solução e apresente o respectivo código para que o objecto remoto passe a ter um *sponsor*.

[3 val] Num contexto .NET Remoting, considere que existem N servidores que disponibilizam objectos de acesso remoto (<http://máquina{1...N}:5050/RemoteExp.soap>) capazes de calcular expressões matemáticas complexas e morosas em tempo de processamento do tipo **Exp**. Apresente uma estratégia e os troços de código que ache necessários, para que um cliente, recorrendo aos N servidores, possa de forma eficiente calcular expressões do tipo **Exp₁+...+Exp_i+...+Exp_n**.

WEB SERVICES

[2 val] Considere a lista de Acções/Actividades (a,b,c,...) que ocorrem desde a criação até à utilização por um cliente de um Web Service (WS). Apresente, justificando, essa mesma lista de Acções/Actividades, ordenada por ordem temporal da sua ocorrência.

a) Criar Cliente	e) Obter ficheiro WSDL	i) Proxy envia mensagem SOAP
b) Criar classe Web Service	f) Criar Proxy classe	j) Criar/Gerar o ficheiro WSDL
c) Definir Interface (operações) do WS	g) Cliente acede ao Proxy	k) O Proxy devolve resposta ao cliente
d) O WS obtém mensagem SOAP	h) O WS devolve mensagem SOAP	l) A operação/método é executada no WS

[3 val] Os Web Services são implementados por uma classe que não mantém estado entre invocações dos seus métodos. No entanto em muitas situações é necessário manter estado no Web Service, entre invocação de métodos e mesmo entre diferentes clientes. Caracterize essas situações, num contexto de utilização da plataforma .NET, apresentando as diferentes formas, indicando vantagens e desvantagens, de manter estado num Web Service.

Com o objectivo de reduzir custos o estado português decidiu que a sua central de compras seria baseada num Web Service (WSCentral), onde todas as encomendas dos organismos públicos teriam de ser efectuadas. Por outro lado obrigou a que todos os candidatos a fornecedores do estado tivessem também um Web Service (WSFornecedor) que teria, entre outros, as operações: obter preço de um produto e comprar produto. O WSCentral perante um pedido de compra de um organismo publico pretende fazer a compra ao fornecedor que apresente o melhor preço.

a)[2 val] Apresente uma possível arquitectura do sistema, indicando as interfaces (operações), interacções entre o WSCentral e os vários WSFornecedor, existência ou não de estados ou outras considerações que ache pertinentes

b)[3 val] Implemente o troço de código da operação do WSCentral que obtém o melhor preço de um produto.

[2 val] As arquitecturas de sistemas distribuídos envolvendo o paradigma WebServices apresentam vantagens quando se pretende o acesso a serviços entre organizações diferentes. Comente e justifique o potencial dos WebServices neste contexto.

2) Considere que um banco pretende implementar um WebService em tecnologias .NET o qual, entre outras, disponibiliza uma operação que permite obter o valor em Euros de uma determinada quantia de uma outra moeda, por exemplo dollars.

- a) [3 val] Apresente a classe C# que implementa esse WebService, a sequência de acções a efectuar pelo programador e o troço de código da aplicação cliente que permite obter valores de câmbio através do WebService do Banco.
- b) [2 val] Considerando que o WebService necessita de autenticação de utilizadores, proponha as diferentes formas de a implementar, ilustrando, para cada uma das formas, as alterações que teria de efectuar na classe C# que implementa o WebService do Banco.

[3 val] Considere que a central de compras do Estado obrigou todos os seus fornecedores a implementar um WebService, em tecnologia .NET, com a mesma interface descrita por um ficheiro WSDL que é passível de ser obtido, por *download*, no site do Ministério das Finanças. Nessa interface, existe uma operação **ObterPrecoProduto** na qual, dado um **Produto** como argumento devolve o seu preço.

Apresente as acções a efectuar, bem como o código de um cliente em modo consola no qual, dado como argumento o URL do WebService de um fornecedor, escreve em *output* o preço de um produto com a designação "Caixa de CD-R80 48x 50 unidades".

[2 val] Para realizar um cliente que aceda a um Web Service, implica conhecer o URL do mesmo, por exemplo, (<http://SomeServer/SomeService.aspx>). Indique e justifique os passos necessários na plataforma .NET para a concretização de um cliente que acede ao Web Service.

[3 val] Considere um Web Service .NET que implementa uma calculadora com operações aritméticas simples. e ainda duas operações que permitem salvar e repor valores em várias memórias globais a todos os clientes. **void SaveValor("M1", valor) - int LoadValor("M1", valor)**

Apresente um esboço da classe que implemente o Web Service com especial ênfase dos serviços de manipulação de valores em memória.

[3 val] Por definição os Web Services não mantêm estado entre chamadas a métodos, dificultando assim a semântica de chamada de um serviço no sentido em que o mesmo não detecta eventuais repetições do mesmo pedido por parte de um cliente. Tirando partido dos mecanismos disponíveis na infra-estrutura .NET, apresente uma estratégia que permita a um Web Service identificar a eventual repetição de pedidos.

[2 val] O Instituto Politécnico de Lisboa é formado por várias escolas superiores, entre as quais o ISEL. Cada escola tem uma biblioteca informatizada de forma independente. Proponha uma solução para a criação de um sistema que permita fazer consultas numa biblioteca virtual do IPL (pesquisa global em todas as bibliotecas de forma transparente para os utilizadores).

[3 val] Considere um Web Service .NET que permite obter o câmbio diário do Dollar através da operação

`double getCambioDollar()` a qual desencadeia uma consulta à Base de Dados do Banco para devolver a taxa de câmbio desse dia. No entanto, verifica-se que a resposta é muito lenta. Assumindo que a taxa de câmbio não muda durante o dia, proponha alterações ao Web Service de forma a tornar mais rápida a execução da operação. Apresente os troços de código que ache relevantes.

[3 val] Considere que pretende implementar um Web Service que permite obter os resultados do sorteio do EuroMilhões (7 números entre 1 e 50). Por forma a minimizar os acessos, o Web Service só permite, a um determinado utilizador, aceder ao serviço uma vez por dia. Tirando partido dos mecanismos disponíveis na infraestrutura .NET, apresente uma estratégia que permita a um Web Service identificar um utilizador e rejeitar pedidos repetidos efectuados no mesmo dia

[2 val] Considere que tem um Web Service .NET que disponibiliza uma operação *AnyOperation* que tem um comportamento estranho. Com o objectivo de compreender esse comportamento estranho, indique uma estratégia e justifique os passos necessários para permitir inspeccionar todas as mensagens SOAP dirigidas ou devolvidas pela operação *AnyOperation*.

[2 val] Considere que tem de implementar um Web Service com necessidade de manutenção de estado entre chamadas de diferentes clientes. Apresente as diversas formas de implementação de manutenção de estado, possíveis de utilizar neste caso, discutindo as vantagens/desvantagens em termos de desempenho, tolerância a falhas e heterogeneidade de possíveis clientes.

[2 val] Na definição do *XML schema SOAP* existe a possibilidade de representar mensagens de erro ou excepção `<SOAP fault>`. Descreva como os Web Services .NET utilizam este mecanismo para retornar excepções específicas de um determinado serviço.

Tendo como referência o ficheiro WSDL (Web Service Definition Language) abaixo apresentado, responda às seguintes questões:

a)[1.5 val] Considerando que o ficheiro WSDL lhe é enviado por mail, descreva os passos a realizar para poder criar uma aplicação que acede ao Web Service.

b)[3.5 val] Apresente o esboço em C# de implementação em .NET do Web Service descrito no ficheiro WSDL, bem como o troço de código de uma aplicação cliente que acede ao Web Service para invocar as operações que ele disponibiliza.

SISTEMAS DISTRIBUÍDOS

Exemplos de Questões para Exame

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions

  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="http://deetc.isel.ipl.pt/ASDexame"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://deetc.isel.ipl.pt/ASDexame" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://deetc.isel.ipl.pt/ASDexame">
      <s:element name="contactByName">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="nome" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="contactByNameResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="contactByNameResult" type="tns:Contacto" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="Contacto">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="telPessoal" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="telExtensao" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="email" type="s:string" />
        </s:sequence>
      </s:complexType>
      <s:element name="listFuncionarios">
        <s:complexType />
      </s:element>
      <s:element name="listFuncionariosResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="listFuncionariosResult" type="tns:ArrayOfFuncionario" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfFuncionario">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Funcionario" nillable="true" type="tns:Funcionario" />
        </s:sequence>
      </s:complexType>
      <s:complexType name="Funcionario">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="numPasta" type="s:int" />
          <s:element minOccurs="0" maxOccurs="1" name="nome" type="s:string" />
          <s:element minOccurs="0" maxOccurs="1" name="contFunc" type="tns:Contacto" />
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>
```

■ ■ ■ (continua no verso)

... (continuação)

```
<wsdl:message name="contactByNameSoapIn">
  <wsdl:part name="parameters" element="tns:contactByName" />
</wsdl:message>
<wsdl:message name="contactByNameSoapOut">
  <wsdl:part name="parameters" element="tns:contactByNameResponse" />
</wsdl:message>
<wsdl:message name="listFuncionariosSoapIn">
  <wsdl:part name="parameters" element="tns:listFuncionarios" />
</wsdl:message>
<wsdl:message name="listFuncionariosSoapOut">
  <wsdl:part name="parameters" element="tns:listFuncionariosResponse" />
</wsdl:message>

<wsdl:portType name="ContactosSoap">
  <wsdl:operation name="contactByName">
    <wsdl:input message="tns:contactByNameSoapIn" />
    <wsdl:output message="tns:contactByNameSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="listFuncionarios">
    <wsdl:input message="tns:listFuncionariosSoapIn" />
    <wsdl:output message="tns:listFuncionariosSoapOut" />
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="ContactosSoap" type="tns:ContactosSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" />
  <wsdl:operation name="contactByName">
    <soap:operation soapAction="http://deetc.isel.ipl.pt/ASDexame/contactByName" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="listFuncionarios">
    <soap:operation soapAction="http://deetc.isel.ipl.pt/ASDexame/listFuncionarios" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

<wsdl:service name="Contactos">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">Gestao Contactos</documentation>
  <wsdl:port name="ContactosSoap" binding="tns:ContactosSoap">
    <soap:address location="http://localhost/1epocaASDi06WS/Contactos.asmx" />
  </wsdl:port>
</wsdl:service>

</wsdl:definitions>
```