

# LEIC/LETI, 2014/15, Repescagem do 1º Teste de Sistemas Distribuídos

## 30 de Junho de 2015

Responda no enunciado, apenas no espaço fornecido. Identifique todas as folhas. Duração: 1h30m

### Grupo I [7v]

Considere o seguinte programa servidor, que funcionará sobre o SUN RPC. Este servidor armazena pares chave-valor (*key-value*), que podem ser escritos ou lidos através das funções remotas apresentadas abaixo.

```
struct store_args {
    char *key;
    char *val;
};
typedef struct store_args store_args;

void *store_1_svc(store_args *argp, struct svc_req *rqstp)
{
    static char * result;
    putInLocalStorage(argp->key, argp->val) ;
    return (void *) &result;
}

char **
load_1_svc(char **argp, struct svc_req *rqstp)
{
    static char * result;
    result = getFromLocalStorage(*argp);
    return &result;
}
```

1. [1,0] Apresente o conteúdo do IDL relativo a este programa. Assuma PROGRAM=62015 e VERSION=1.

2. [0,8] Uma hipotética alternativa a ter descrito a interface remota do serviço no IDL anterior teria sido definir os tipos de dados e assinaturas das funções num simples ficheiro .h. Apresente um exemplo da sua resposta anterior que ilustrem porque é que o .h não é solução.

Caso não tenha respondido à alínea anterior, apresente um exemplo livre (cotação limitada a 70% neste caso).

3. [1,2] Considere o seguinte esboço de um programa cliente em SUN RPC, ainda em construção, que invoca o serviço anterior:

```
int main(...) {
    char *load_res;
    store_args args;

    /* We copy the contents of key1 to key2 */
    load_res = load_1("key1", NULL);

    args.key = "key2";
    args.val = load_res;
    store_1(&args, NULL);

    return 0;
}
```

Ajude o programador, indicando-lhe 2 aspetos de programação de SUN RPC que são fundamentais para o programa funcionar em SUN RPC. Proponha as alterações necessárias.


4. O estabelecimento da sessão (ou *binding*) é uma etapa fundamental de qualquer programa cliente em RPC. No caso do Sun RPC, indique:

a. [0,7] Quais os principais argumentos que a função que estabelece a sessão recebe.


b. [0,8] Os passos que são tomados até a sessão (*binding*) ser estabelecida com sucesso.


5. O servidor acima, quando usado sobre UDP, oferece a semântica pelo-menos-1-vez.

a) [0,8] Pode uma invocação nesta semântica devolver “erro de RPC” devido a faltas na comunicação ou no servidor? Se sim, apresente num diagrama um exemplo em que uma invocação do cliente devolve “erro de RPC”. Se não, justifique.

Assuma que a rede não é fiável e que o nó servidor pode falhar.


b) [0,7v] Para esta aplicação, seria vantajoso mudar para no-máximo-1-vez?


6. [1,0] Considere o módulo de comunicação de um servidor RPC que oferece a semântica no-máximo-1-vez.

Apresente a sequência de passos (pode ser pseudo-código) que descreve o que esse módulo faz quando recebe um pedido vindo do cliente.




3. Indique se cada afirmação seguinte sobre “garbage collection” é verdadeira ou falsa e justifique.

a) [0,8] “O uso de leases implica cliente e servidor com relógios sincronizados.”


b) [0,8] “Em Java RMI, um objeto remoto é libertado assim que já não tenha referências locais a apontar para ele; independentemente do número de referências remotas que tenha.”


### Grupo III [7,0v]

1. [0,7] Considere o caso de uma empresa que decidiu evoluir a sua infra-estrutura de interoperação utilizando Web Services. A empresa tem já descrita a grande maioria das entidades informacionais como XML schemas em ficheiros .xsd. Poderá reaproveitar este trabalho na definição dos Web Services? Seja claro na resposta relacionando com o documento WSDL e respectivas secções.


2. Considere o extracto de XML schema definido na empresa para normalizar as mensagens de erro.

```
<xsd:complexType name="Fault">
<xsd:sequence><xsd:element minOccurs="0" name="reason" type="xsd:string"/>
</xsd:sequence>
</xsd:complexType
```

[0,7] Explique sucintamente o significado dos principais elementos deste XML schema: name, element, sequence, minOccurs, name e type.


3. Considere o seguinte extrato de WSDL com a definição de um serviço.

```
<wsdl:portType name="Interface">
  <wsdl:operation name="op1">
    <wsdl:input name="op1Request" message="tns:op1RequestMsg"/>
    <wsdl:output name="op1Response" message="tns:op1ResponseMsg"/>
  </wsdl:operation>
</wsdl:portType>
```

- a. [0,7] Explique onde se encontram declarados os parâmetros da operação e os respetivos tipos no documento WSDL.


- b. O que deveria acrescentar para existir a possibilidade do cliente receber uma exceção lançada pelo servidor cujo **formato correspondesse ao formato genérico da alínea 2?**

- i. [0,5] Descreva o XML a acrescentar à secção portType.


- ii. [0,5] Descreva o XML a acrescentar à secção message.


- c. [0,5] Suponha que pretende criar outra operação OP2 com o mesmo tipo de mensagem de erro. **Escolha a opção certa. Resposta errada desconta ¼ da cotação.**

- a. Não é possível, cada operação tem de ter mensagens diferentes.  
b. É possível a reutilização da mensagem de erro na descrição da OP2.  
c. O WSDL não tem mecanismos para reutilização de elementos de diferentes secções.  
d. Se a operação se chamar OP2 a fault message tem de se chamar faultOP2.

- d. Suponha que a empresa usa Kerberos para autenticar os utilizadores e os servidores. Pretende-se incluir o Ticket e o Autenticador na invocação dos web services.

- a. [0,4] Que alterações deve efectuar no WSDL?


- b. [0,4] Em que secção da mensagem SOAP devem ser enviados o Ticket e Autenticador?


- c. [0,5] Que componente do sistema deverá executar a validação do ticket e autenticador do lado do servidor?


e. O Java RMI tem diversos conceitos que usam nomes semelhantes aos da tecnologia de Web Services mas mesmo quando usada com Java-WS diferem substancialmente.

a. [0,7] Em Java RMI existe o conceito de uma **referência remota para um objecto**. Este conceito não existe em Web Services. Em sua opinião qual a razão da sua não existência, justificando com base nos pressupostos que presidiram à criação dos Web Services?


b. Um objecto em Java quando interactua com um objecto remoto fá-lo através de um proxy.

i. [0,7] Indique uma diferença fundamental dos proxies em Web Services relativamente aos proxies de Java RMI quando são invocados por um objecto cliente. Justifique.


ii. [0,7] Indique uma semelhança. Justifique.
