

**Mestrado Integrado em Engenharia Informática**  
**Sistemas Distribuídos – Exame Recurso, 30 de Junho de 2016**  
**2º Semestre, 2015/2016**

**NOTAS:** Leia as questões atentamente antes de responder. **O exame é sem consulta. A duração do exame é 2h30min.** O exame contém **12** páginas.

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

1) Indique se cada afirmação é **[V]erdadeira** ou **[F]alsa (nota: respostas incorretas descontam)**:

- Num sistema distribuído, cada componente do sistema apenas tem uma visão parcial do sistema.
- A retransmissão de mensagens é uma forma de mascarar algumas falhas que ocorrem no sistema de comunicações.
- O programa *wsgen* cria permite criar classes Java para um cliente aceder a um servidor *web services* SOAP.
- Num servidor de *web services* em Java é possível que dois *threads* estejam concorrentemente num dado momento a executar atendendo a pedidos de diferentes clientes.
- Nos *web services* SOAP, os objetos são passados por valor.
- No RMI, a semântica da invocação é pelo menos uma vez / "at least once".
- No RMI, os objetos são passados por valor.
- O endereço IP duma máquina (193.123.14.29) é um nome simbólico.
- Um serviço de nomes permite obter os atributos de uma entidade que satisfaz uma dada descrição.
- Na resolução iterativa de um nome que envolva vários servidores, o cliente é o elemento que realiza mais operações de comunicação.
- Num sistema de disseminação de eventos baseado no conteúdo, é possível que as mensagens sejam opacas para o sistema, i.e., o seu conteúdo não pode ser conhecido pelo sistema.

2) Considere o código do servidor REST apresentado no Anexo A, o qual foi escrito recorrendo à biblioteca Jersey. Este servidor oferece uma API parcial do serviço Imgur que, como sabe, permite o alojamento de imagens organizadas em álbuns.

- a) Considerando a listagem de anotações que se encontram no final do Anexo A, complete o código fornecido do servidor, como achar necessário. É livre de escolher os caminhos (paths) que identificam os recursos manipulados pela API.
- b) Suponha que se pretende fazer uma versão SOAP do servidor que irá executar **noutro processo, mas na mesma máquina**. Por simplicidade, os métodos da versão SOAP serão implementados invocando o servidor REST. Complete o código do ANEXO B do servidor SOAP.

- 3) O sistema Doodle.com é um serviço online, gratuito, que permite agendar um evento para uma data e/ou horário mais conveniente para um grupo de utilizadores, onde a escolha da melhor data e horário é feita de forma colaborativa. Uma vez criado o *doodle* para o evento, onde constam uma série de datas e horários alternativos, os utilizadores são chamados a indicar as suas preferências e impedimentos.

7 de Julho de 2016					
	9:00	11:00	2:00	4:00	8:00
Tom	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Paula	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Chris	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="text"/>	<input type="checkbox"/>				

Um *doodle* consiste numa página web, como a apresentada na figura, onde o utilizador pode ver as datas alternativas existentes e para cada uma destas indicar se lhe agrada ou não. Poderá também ver as respostas já dadas pelos outros utilizadores. Responder a um *doodle* não demora mais do que uns minutos, tipicamente, e fica concluído premindo um botão que envia as preferências do utilizador para o *backend* do serviço.

Quando todos responderam, o responsável pelo *doodle*, com base em todas as respostas dadas, pode seleccionar a melhor alternativa, por exemplo, aquela para a qual ninguém indica um impedimento ou a data que maximiza o número de presenças.

Complementarmente à ideia geral descrita acima, o sistema apresenta as seguintes características:

- Um *doodle* é identificado pelo URL da página web onde está alojado.
- Os utilizadores do sistema são identificados por um endereço de email.
- Qualquer utilizador com um endereço de email válido, em qualquer parte do mundo, pode criar um *doodle* e convidar utilizadores para participarem (via email).
- O serviço revelou-se muito popular em todo o mundo, com versões nas mais diversas línguas.

Com base na descrição acima, responda às seguintes questões, **RISQUE** a opção que não interessa e **JUSTIFIQUE** a sua resposta com argumentos tão específicos quanto o possível face ao cenário proposto.

- a) A infraestrutura de suporte do serviço terá benefícios em recorrer ao particionamento da informação nos servidores.

**Sim / Não**, porque ...

- b) A infraestrutura de suporte do serviço terá benefícios em recorrer à replicação da informação nos servidores.

**Sim / Não, porque ...**

- c) Assumindo que na conceção do sistema se optou por **não utilizar** geo-replicação, indique qual o impacto esperado dessa decisão nas seguintes propriedades e que características do sistema poderão justificar tal decisão.

Custo:

Latência:

Tolerância a falhas.

- d) Suponha que o Doodle.com pretende permitir que outras aplicações possam aceder e modificar a informação dos *doodles*. Para tal, pretende disponibilizar quatro operações: (1) criar *doodle*, (2) apagar *doodle*, (3) consultar o estado dum *doodle*, (4) modificar conteúdo dum *doodle*. Considerando que a operação de consultar um *doodle* é invocada muito mais vezes que todas as outras operações, implementaria este serviço usando REST ou SOAP? Justifique.

**REST, porque... / SOAP, porque...**

e) Suponha que os utilizadores que já responderam ao *doodle* pretendem ser notificados sempre que um novo utilizador expressa as suas opções. A utilização dum sistema de *middleware publish/subscribe* (editor/assinante) seria interessante para endereçar este requisito? Se sim, explique como usaria o sistema (tipo de sistema de disseminação, quando é feita a subscrição, quando é feita a publicação e qual o conteúdo do evento disseminado). Se não, justifique porque razão não é interessante.

4) Considere um sistema de invocação remota de métodos em que os objetos eram passados por referência. Neste caso, seja X uma *hashtable* passada por referência. Discuta o que acontece quando se invoca, no servidor, o método *clear* no objeto X (recebido do cliente). NOTA: na sua resposta deve explicar o efeito no cliente e no servidor, e como é que este efeito é alcançado.

Nome: \_\_\_\_\_ Número: \_\_\_\_\_

5) Considere o seguinte protocolo em que a Alice (A) envia uma sequência de 3 mensagens M1, M2, e M3 para o Bob (B). À partida, a Alice e o Bob já conhecem as chaves públicas um do outro ( $K_{Apub}$  e  $K_{Bpub}$ , respetivamente).

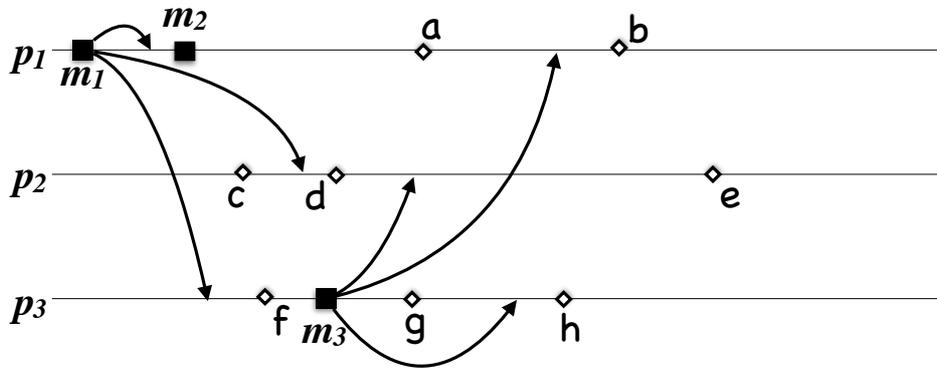
- 1) A -> B:  $\{A, Na\} K_{Bpub}$
- 2) B -> A:  $\{Na, Nb, Ks\} K_{Apub}$
- 3) A -> B:  $\{M1\}Ks, H(M1+Nb)$
- 4) A -> B:  $\{M2\}Ks, H(M2+M1)$
- 5) A -> B:  $\{M3\}Ks, H(M3+M2)$
- 6) B -> A:  $H(M3+M2+M1+Nb)$

a) Tendo em conta o protocolo anterior, preencha a tabela seguinte, colocando X onde apropriado. A colocação de X numa dada célula deverá indicar que a componente da mensagem indicada nessa linha **contribui** para a garantia associada à respetiva coluna. Por exemplo, como indicado na tabela,  $\{M2\}Ks$  na mensagem enviada no passo 4) do protocolo contribui para a garantia de confidencialidade das mensagens a trocar.

Garantias \ Componentes das mensagens	Confidencialidade	Autenticação	Integridade	Não repetição (Non-Replaying)	Deteção da supressão de mensagens
1) ... Na ...					
2) $\{Na, Nb, Ks\} K_{Apub}$					
3) $H(M1+Nb)$					
4) $\{M2\} Ks$	<b>X</b>				
5) $H(M3+M2)$					

b) Qual o objetivo da mensagem enviada pelo Bob à Alice no passo 6?

6) Considere o seguinte diagrama que ilustra um padrão de comunicação em grupo, envolvendo três processos:  $p_1$ ,  $p_2$  e  $p_3$ ; e as mensagens:  $m_1$ ,  $m_2$  e  $m_3$ . O momento de entrega das mensagens  $m_1$  e  $m_3$  é fixo e indicado por setas a cheio. Para  $m_2$ , o momento de entrega dessa mensagem em cada um dos 3 processos, tem diversas alternativas que foram identificadas por letras. Por exemplo,  $f$  corresponde à entrega de  $m_2$  em  $p_3$  num momento anterior ao envio de  $m_3$ , enquanto  $g$  (ou  $h$ ) ocorre após esse envio. Note que em cada processo,  $m_2$  será entregue uma e só uma vez.



a) Indique se é verdadeiro (V) ou falso (F) cada uma das seguintes afirmações. Respostas incorretas descontam.

\_\_\_ Para o par  $(m_1, m_2)$ , se  $\{a, c, f\}$  corresponderem à entrega da mensagem  $m_2$ , então a entrega destas mensagens é compatível com uma **ordenação FIFO**.

\_\_\_ Para o trio  $(m_1, m_2, m_3)$ , se  $\{b, e, h\}$  corresponderem à entrega da mensagem  $m_2$ , então a entrega de todas as mensagens é incompatível com uma **ordenação total**.

\_\_\_ Para o par  $(m_2, m_3)$ , se  $g$  corresponder à entrega da mensagem  $m_2$  em  $p_3$ , esta entrega é compatível com uma **ordenação causal**.

b) Considerando conjuntamente as mensagens  $m_1$ ,  $m_2$  e  $m_3$ , indique um conjunto de eventos que satisfaça os seguintes requisitos:

{ } é compatível com uma ordenação **FIFO**

{ } é compatível com uma ordenação **causal**

{ } é compatível com uma ordenação **total causal**

{ } é compatível com uma ordenação **não total**.



8) Considere um sistema de replicação primário-secundário, com 1 primário e 5 secundários.

a) Caso se pretenda tolerar a falha de 3 servidores, o primário pode responder aos clientes após receber a confirmação (ack) de quantos secundários? Justifique.

b) Explique porque razão pode ser importante o cliente guardar o identificador da última versão observada e enviar este identificador quando executa uma operação de leitura.

<b>Nome:</b> _____	<b>Número:</b> _____
--------------------	----------------------

Página em branco ...

## ANEXO A

```
...  
import common.Album;  
import common.Image;
```

```
public class ImgurRestServer {  
    public static final int PORT = 8080;
```

```
public void createAlbum( Album album ) {  
    //cria um novo album..  
}
```

```
public void updateAlbum(  String id, Album data ) {  
    //atualiza informação dum album (e.g. novo nome)  
}
```

```
public void deleteAlbum(  String id) {  
    //apaga um dado album  
}
```

```
public List<Image> getAlbumImages(  String id) {  
    //obtem as imagens de um dado album  
}
```

```
public Image getAlbumImage(  String album,  
 String image) {  
    //obtem uma imagem de um dado album  
}
```

```
public static void main(String[] args ) throws Exception {
    URI baseUri = UriBuilder.fromUri("http://0.0.0.0/").port(PORT).build();
    ResourceConfig config = new ResourceConfig();
    config.register(new ImgurRestServer());
    JdkHttpServerFactory.createHttpServer(baseUri, config);
    System.err.println("Imgur REST Server ready... ");
}
}
```

**Anotações JAVA REST Jersey:**

- @Path( )
- @GET
- @POST
- @PUT
- @DELETE
- @Consumes( )
- @Produces( )
- @PathParam( )

**LListagem de Media Types a considerar:**

- MediaType.APPLICATION\_OCTET\_STREAM
- MediaType.APPLICATION\_JSON

## ANEXO B

```
import ...
```

```
public class ImgurSoapServer {  
    public static final int PORT = 9090;
```

```
public Image getAlbumImage( String album, String image) {  
    ClientConfig config = new ClientConfig();  
    Client client = ClientBuilder.newClient(config);  
  
    WebTarget target = client.target(" ");  
  
    Image img = target.path(" ")  
        .request()  
        .accept(MediaType. )  
        .get(Image.class);  
  
    return img;  
}  
  
...  
  
public static void main(String[] args ) throws Exception {  
    Endpoint.publish("http://0.0.0.0:" + PORT + "/ImgurSoap", new ImgurSoapServer() );  
    System.err.println("Imgur SOAP Server ready... ");  
}  
}
```