FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Departamento de Informática

Mestrado Integrado em Engenharia Informática
Sistemas Distribuídos – 1º teste, 11 de Abril de 2014 - versão B
1º Semestre, 2014/2015

NOTAS: Leia as questões atentamente antes de responder. **O teste é sem consulta. A duração do teste é 1h30min**. O teste contém **5** páginas.

Nome: _____ **Número**: _____

1) Considere o contexto do trabalho prático, em que um conjunto de servidores fornecem acesso a ficheiros através de RMI e Web Services (SOAP) e em que um servidor de contacto mantém informação sobre os nomes que cada servidor tem (como no trabalho prático, pode haver mais do que um servidor com um dado nome). Complete o código do anexo A, um excerto do código do sistema com o código adequado face aos comentários no código.

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

___ Na rede *peer-to-peer* eDonkey/eMule as pesquisas dum *peer* são propagadas para os outros *peers*.

___ No sistema *peer-to-peer* hierárquico KaZaA/Skype as pesquisas dum *peer* são propagadas para os outros *peers*.

___ Numa rede *peer-to-peer* estruturada (DHT) com *N* nós, se cada peer conhecer apenas dois outros nós, uma pesquisa tem de contactar em média aproximadamente *N/4* nós.

___ Um sistema cliente/servidor tem uma baixa tolerância a falhas.

___ Num sistema cliente/servidor replicado com 2 servidores, seja *r* o número de pedidos de leitura e *w* o número de pedidos de escrita, em média, cada servidor tem de atender *(r + w) / 2* pedidos.

___ Num sistema distribuído existe paralelismo real dos diferentes processos (clientes e servidores) que participam no sistema.

___ As máquina virtuais do tipo aplicação (e.g. Java VM) simplificam a criação de sistemas distribuídos.

___ Os ataques de "*denial of service*" podem ser resolvidos facilmente usando técnicas criptográficas.

___ Quando se descarta uma mensagem com um *checksum* incorreto sem tomar mais nenhuma ação, transforma-se uma falha arbitrária numa falha por omissão.

___ O tempo de recuperação de uma falha é muito importante para a fiabilidade dum sistema distribuído.

No contexto da invocação remota de métodos/procedimentos:

___ Os *web services* (SOAP), por omissão, implementam uma semântica de invocação "*at least once*".

___ Para implementar uma semântica "*exactly-once*" é necessário manter estado em memória estável.

___ O WSDL inclui informação sobre o endereço que deve ser contactado para invocar uma operação no servidor de *web services*.

___ O servidor *rmiregistry* permite manter o registo de dois servidores com o mesmo nome.

___ A ativação de objetos remotos é interessante em situações em que apenas existe um pequeno número de servidores, cada um utilizando poucos recursos.

___ Se os objetos Java fossem serializáveis por omissão, isso colocaria problemas de segurança quando usados no contexto do Java RMI.

3) Considere que pretende implementar um sistema semelhante ao *Spotify*, que permite aos utilizadores acederem a músicas em formato digital, com as seguintes características: (i) o serviço armazena milhões de músicas; (ii) a popularidade das músicas segue uma distribuição Zipf, ou seja, existe um pequeno conjunto de músicas muito populares e um número muito elevado de músicas pouco populares; (iii) a popularidade das músicas varia consoante o país.
Considere que pretende apenas tratar da parte do acesso às músicas e que tem à sua disposição três centros de dados, um em cada um dos seguintes continentes: Europa, Ásia, América.

   a) Proponha uma arquitetura para implementar este serviço. Justifique a sua opção, indicando como distribuiria as músicas pelos três centros de dados e quais as vantagens da sua solução face a outras possíveis soluções.

   b) Seria interessante estender a sua arquitetura usando servidores presentes nos ISPs (e.g. rede Akamai)? Justifique explicando a razão da sua resposta.

Sim/ Não , porque…

4) Comente a seguinte afirmação "A utilização dum modelo de comunicação reativa (e.g. *publish-subscribe*) é interessante para a distribuição de informações sobre os eventos importantes num jogo de futebol".

Sim/ Não , porque...

5) Explique porque motivo é interessante colocar um *proxy* próximo dum servidor web que gera páginas dinâmicas e quais os problemas/desafios que essa solução tem.

6) No Java RMI não é possível migrar um servidor, i.e., mudar a máquina em que um servidor executa. Discuta as modificações que seria necessário efetuar para permitir essa funcionalidade.
Sugestão: comece por considerar as referências que os clientes mantêm para os servidores e a partir daí explique todas as alterações que seria necessárias.

ANNEX A

```
/** Exceção indicando que ficheiro/diretório não existe. */
public class NotFoundException extends Exception { ... }

/** Classe com informação sobre ficheiro. */
public class FileInfo implements                    {
    public String name;
    public long length;

    public FileInfo( String name, long length) {
        this.name = name;
        this.length = length;
    }
}

/** Interface do servidor de contacto RMI. */
public interface IContactServer extends Remote                    {
/* Devolve lista com URLs dos servidores com o nome name ou null caso não exista nenhum servidor. */
    public String[] listServers( String name) throws
}

/** Interface do servidor de ficheiros RMI. */
public interface IFileServer extends Remote                    {
/* Devolve informação sobre ficheiro path. Em caso de erro lança NotFoundException. */
    public FileInfo getAttr ( String path) throws NotFoundException, ... ;
}

/** Classe do servidor de ficheiros RMI. */
public class FileServer extends                    implements IFileServer { ... }

/** Classe do servidor de ficheiros web service SOAP. */

public class FileServerWS {

    public void FileInfo getAttr ( String path) throws                    {...}
    ...
    }

/** Classe principal do cliente */
public class FileClient {
    private IContactServer cserver;    /* referência para o servidor de contacto */
    /** Devolve referência para servidor RMI. Em caso de erro devolve null. */
    public IFileServer getRMIRef ( String url) {

    }
    /** Devolve referência para servidor web services - considere que existe uma classe "service" com o
    nome FileServerWSService e com um construtor com apenas um parâmetro que é o URL. */
    public ws.FileServerWS getWSRef ( String url) {

    }
```

```
/** Devolve informação sobre o ficheiro path que está num servidor com o nome serverName.
* Suponha que todos os servidores com o nome serverName contêm os mesmos ficheiros.
* Devolve null caso o ficheiro não seja encontrado. */
protected FileInfo getAttr ( String serverName, String path) {
```

```
} }
```