**Exame Normal 2023/24**

**1.** Relativamente à passagem de argumentos em chamadas de métodos remotos usando Java RMI,

A. objetos de classes que implementem java.rmi.Remote são passados por valor e tipos primitivos são passados por valor.

B. objetos de classes que implementem java.rmi.Remote são passados por valor e tipos primitivos são passados por referência.

C. objetos de classes que implementem java.rmi.Remote são passados por referência e tipos primitivos são passados por valor.

D. objetos de classes que implementem java.rmi.Remote são passados por referência e tipos primitivos são passados por referência.

**2.** Relativamente à comunicação de utilizando IP multicast é verdadeiro afirmar que

é a C não a D

A. o emissor envia mensagens individuais a cada destinatário separadamente e a fiabilidade da comunicação é garantida por um mecanismo de re-transmissão.

B. o emissor envia mensagens individuais a cada destinatário separadamente e não há garantias de fiabilidade na entrega de mensagens.

C. o emissor envia mensagens para um grupo específico de recetores e não há garantias de fiabilidade na entrega de mensagens.

D. o emissor envia mensagens para um grupo específico de recetores e a fiabilidade da comunicação é garantida por um mecanismo de re-transmissão.

**3.** Na construção de sistemas de invocação remota, o protocolo request-reply-acknowledge reply (RRA) pode ser usado para se conseguir a semântica at-most-once. Para o servidor re-enviar respostas sem re-executar operações

A. o histórico é um registo de respostas previamente transmitidas pelo servidor que nunca podem ser apagadas.

B. o histórico é um registo de pedidos previamente recebidos pelo cliente que nunca podem ser apagados.

C. o histórico é um registo de pedidos previamente recebidos pelo cliente que podem ser apagados quando a respetiva mensagem de acknowledge reply for recebida.

D. o histórico é um registo de respostas previamente transmitidas pelo servidor que podem ser apagadas quando a respetiva mensagem de acknowledge reply for recebida.

**4.** Em relação às ligações HTTP persistentes (ligações keep-alive), que permitem a uma única ligação TCP manter-se aberta durante algum tempo, para múltiplos pedidos e respostas HTTP, podemos afirmar que

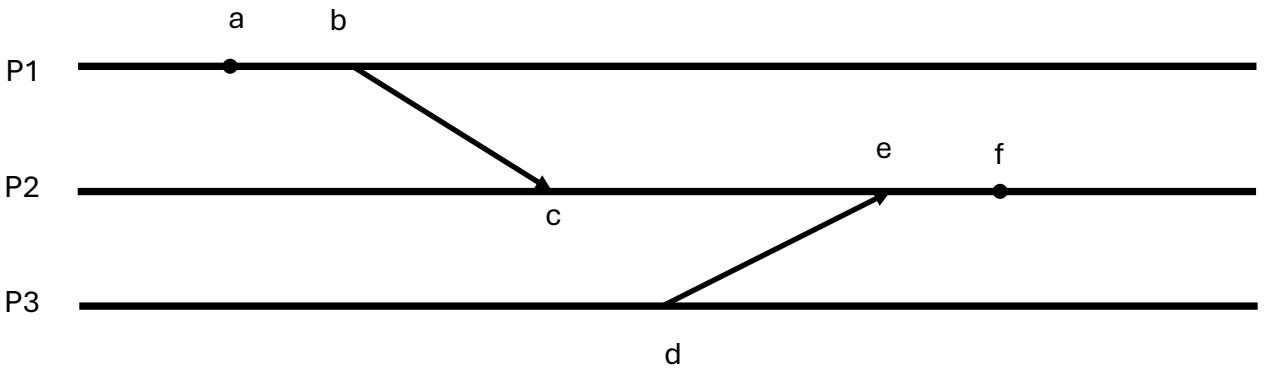
A. um tempo de keep-alive muito reduzido obriga a re-estabelecer ligações TCP frequentemente e um tempo de keep-alive muito elevado resulta na ocupação excessiva de recursos do servidor.

B. um tempo de keep-alive muito reduzido não obriga a re-estabelecer ligações TCP frequentemente e um tempo de keep-alive muito elevado não resulta na ocupação excessiva de recursos do servidor.

C. um tempo de keep-alive muito reduzido obriga a re-estabelecer ligações TCP frequentemente e um tempo de keep-alive muito elevado não resulta na ocupação excessiva de recursos do servidor.

D. um tempo de keep-alive muito reduzido não obriga a re-estabelecer ligações TCP frequentemente e um tempo de keep-alive muito elevado resulta na ocupação excessiva de recursos do servidor.

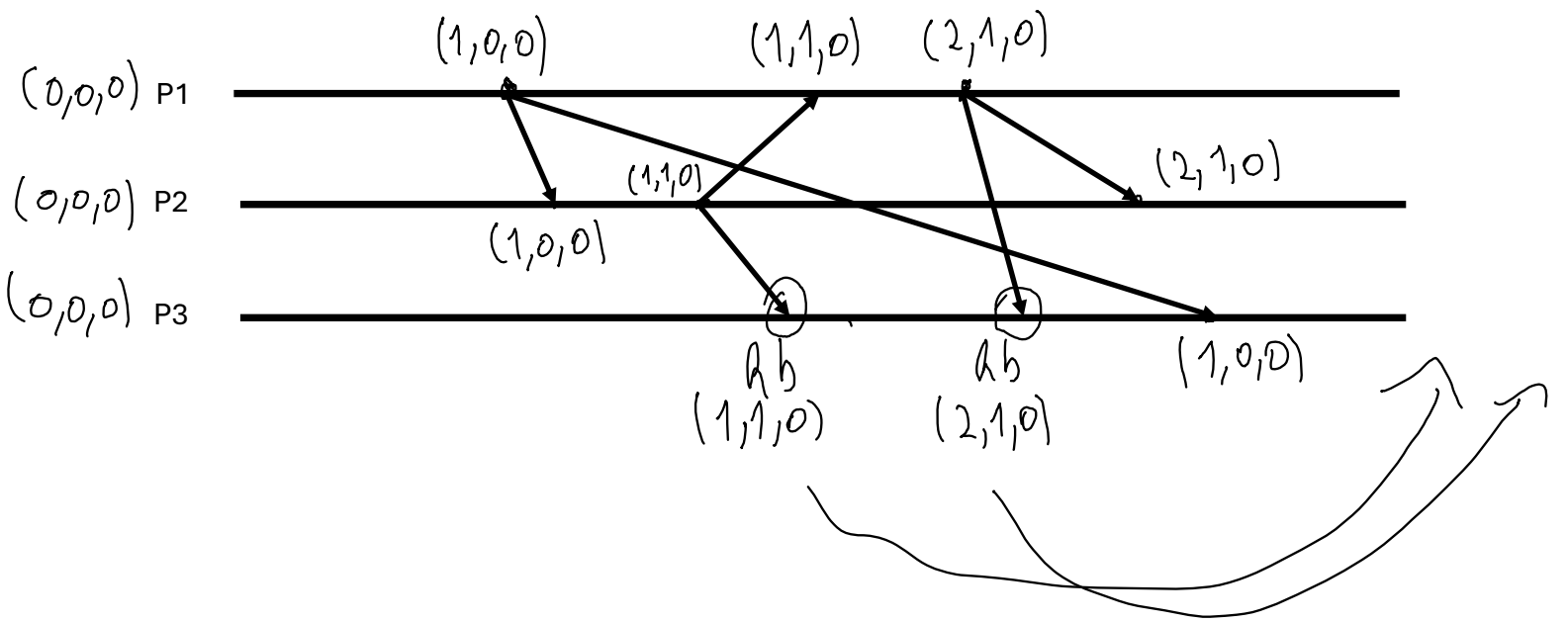
**5.** Considere a troca de mensagens entre os processos P1, P2 e P3 representada na figura que se segue. Trata-se de eventos de envio e receção de mensagens ponto-a-ponto e eventos locais.




**5.a)** Identifique dois eventos x e y tais que a → y (o símbolo → significa "happened before"). Justifique.

a → b. Vetor a < b

**6.** Considere a troca de mensagens multicast entre os processos P1, P2 e P3 representada na figura seguinte. Usando o algoritmo de vector clocks para garantir ordenamento causal em comunicação multicast, indique na figura os vector timestamps em cada ponto de envio e de receção, bem como nas próprias mensagens.Contabilize apenas os eventos de envio de mensagens. Indique eventuais receções de mensagens multicast que sejam colocadas na hold-back queue e o ponto em que são entregues à aplicação.



**7.** Considere um cenário no qual a Alice (A) envia um documento D muito grande ao Bob (B). As chaves públicas de ambos são conhecidas por todos previamente. A Alice pretende assegurar a integridade e a confidencialidade do documento e pretende receber uma confirmação de que o Bob recebeu e teve acesso a esse documento. Apresente uma troca de mensagens entre A e B que permita dar essas garantias de
segurança e justifique a resposta.

**8.** Descreva o papel do protocolo OAuth na segurança de páginas e serviços web, como este protocolo melhora a experiência do utilizador e a segurança comparativamente com a utilização de autenticação baseada em senhas.

**9.** A versão inicial da rede Gnutella difere da versão melhorada com a introdução de super-nós (também designados ultrapeers). Essa melhoria exigiu uma grande mudança na forma como os nós publicam conteúdo, isto é, na forma como os nós anunciam os seus ficheiros para que sejam conhecidos pelos outros peers. Em que consistiu essa mudança e por que razão foi necessária?

**10.** Os sistemas de publish-subscribe permitem a realização de comunicação indireta, sendo possível conceber aplicações nas quais os publishers não têm de conhecer a localização dos subscribers (desacoplamento espacial) nem é necessário que estejam ligados ao mesmo tempo (desacoplamento temporal). Como é conseguido esse desacoplamento espacial e temporal?

(7*)
1º Alice cifra uma hash do documento D usando a chave simétrica por ela gerada. Encripta-a a chave simétrica de modo a apenas Bob a conseguir desencriptar

$A \rightarrow B: \{H(D)\}_{Ks}\ \ \{Ks\}_{Kpub B}$

2º Após receber o documento, Bob reenvia a chave privada cifrada com a sua chave privada (assinatura)

$B \rightarrow A: \{Ks\}_{K_{B\_priv}}$

3º Alice recebe a confirmação e encerra

(8) O protocolo OAuth permite que uma aplicação aceda outra sem que precise de dados explícitos (privados) do cliente sobre a app "third party". Funciona através do redirecionamento do user para a app secundária fazendo este o login seguro na mesma sendo posteriormente enviado um token de acesso temporário à app de origem.

(9) Passagem de modelo puramente P2P para modelo de superNós. Porquê? Problema de flooding, largura de banda toda ocupada por queries

(10) Publishers não sabem quem recebe a mensagem
↳ Publicam-nas em tópicos que os clientes subscrevem.
↳ Quem trata da entrega é um middleware.
Desacoplamento temporal:
Armazenamento temporários em filas e quando o cliente subscreve um tópico é recebe-os.