LETI/LEIC 2017/18, Repescagem do 2º Teste de Sistemas Distribuídos

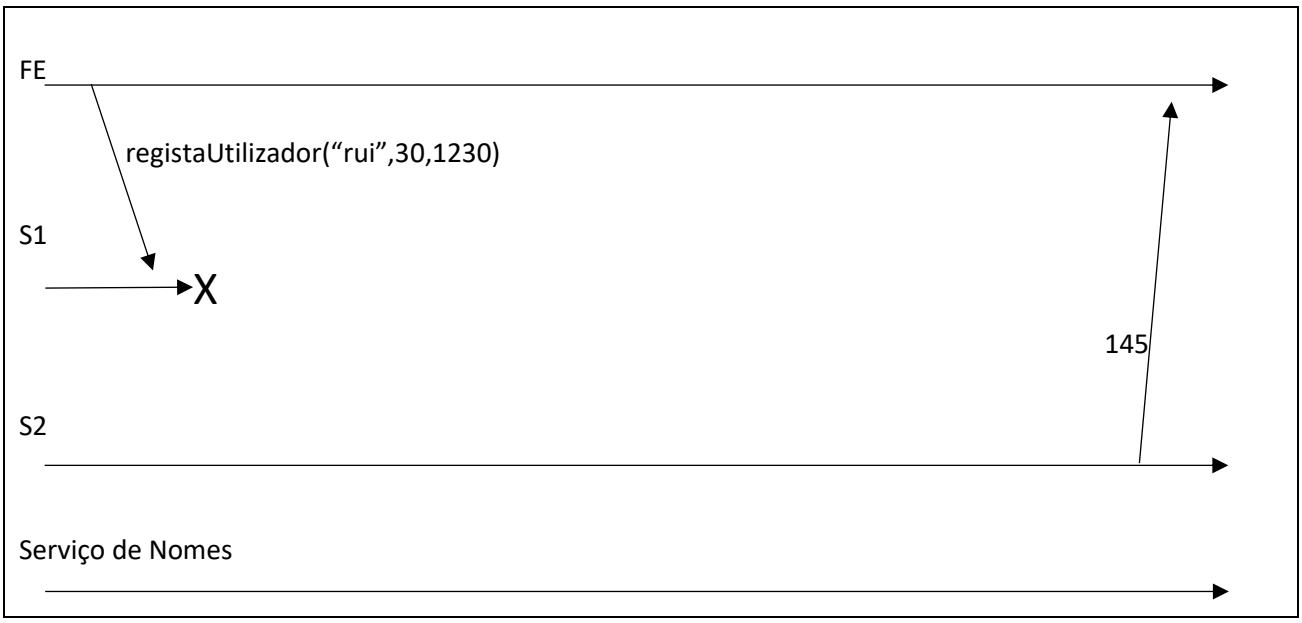
3 de julho de 2018

Responda no enunciado, usando apenas o espaço fornecido. Identifique todas as folhas.
Uma resposta errada numa escolha múltipla desconta 1/N do valor da pergunta em N alternativas.
Duração da prova: 1h30m

Grupo I [7 valores]

1. Considere um sistema replicado com o protocolo *primary-backup* estudado na cadeira, com as seguintes propriedades particulares:
   - Este sistema tem dois servidores (um primário, um secundário), ligados numa rede fiável que garante que o tempo de transmissão de mensagens é sempre inferior a 1 ms.
   - O primário envia provas de vida ao secundário com um período de 20 ms. Assuma que o desvio entre os relógios de ambos os servidores é negligenciável.
   - A informação sobre qual é o primário atual é mantida num serviço de nomes. Quando um novo primário é eleito, o seu registo junto do serviço de nomes demora 3 ms em média (incluindo já o tempo de comunicação).
   - O tempo médio entre falhas de cada servidor é de 100.000 s. Assuma que, sempre que o primário falha, existe um secundário disponível para o substituir.

   a) [1,3v] Considere uma execução em que o *front-end* de um cliente começa por pedir para registar um novo utilizador no sistema e no final recebe o identificador 145 como retorno; entre estes dois momentos, o servidor S1 falha.
   Complete o seguinte diagrama com todas as mensagens que estão em falta.
   Nota: diferentes respostas são possíveis; escolha apenas uma.



b) [0,9v] Apresente uma estimativa do tempo de recuperação médio (MTTR) deste sistema numa situação em que o primário falha e o secundário está disponível para o substituir. Justifique apresentando as fórmulas que levaram à sua resposta.

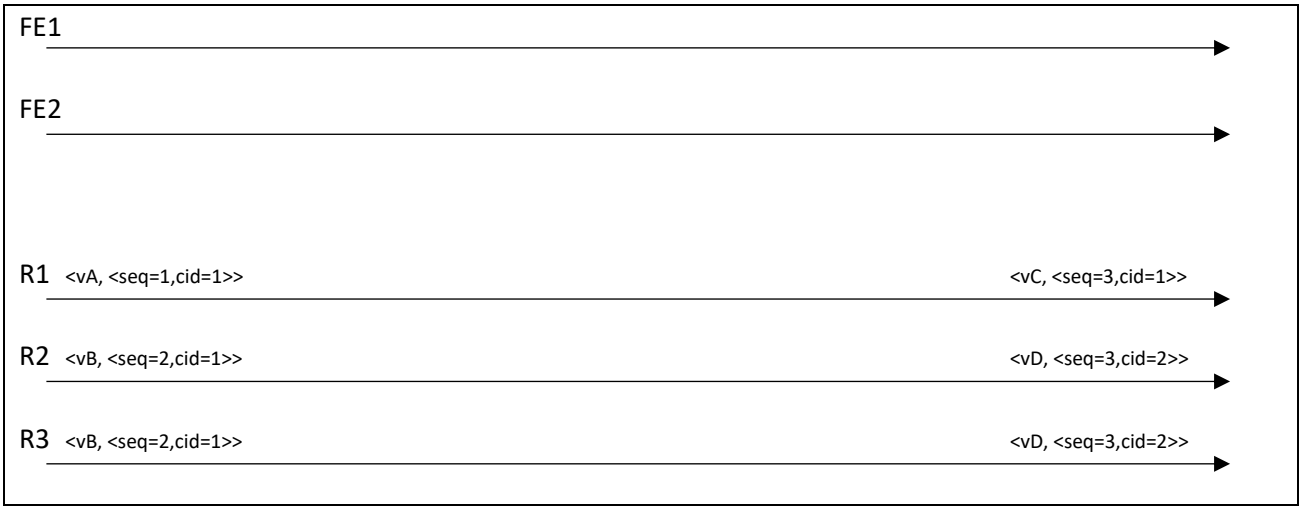
Tempo para secundário detetar falha do primário: P + tmax = 20 + 1 ms = 21 ms
Tempo para secundário se registar como novo primário no serviço de nomes: 3 ms
MTTR = tempo em que o sistema esteve sem servidor primário correto = 21 + 3 ms = 24 ms

Resposta alternativa, também correta: calcular MTTR na perspetiva do front-end (incluir tempo adicional para o FE descobrir e invocar o novo primário)

c) [0,9v] Qual a disponibilidade deste sistema? Não precisa apresentar o valor final, apenas a fórmula com os valores concretos. Caso não tenha respondido à alínea anterior, pode assumir MTTR = 1 s.

2. [1,0v] *"O protocolo primary-backup é sempre melhor que o quorum consensus pois, para tolerar f falhas silenciosas simultâneas, exige menos réplicas no total."*
Contradiga esta afirmação apresentando uma vantagem do protocolo *quorum consensus*.

3. Considere agora um sistema replicado que recorre ao protocolo *quorum consensus* com 3 réplicas. Observou-se o estado das réplicas em dois momentos, que se descrevem na figura abaixo.



a) [0,8v] Caso um cliente leia quando o sistema está no primeiro estado acima, que valor obterá? Justifique.

b) [0,8v] Caso um cliente leia quando o sistema está no último estado acima, que valor obterá? Justifique.

c) [1,3v] Complete o diagrama acima com as mensagens trocadas que levaram o sistema até ao último estado. *Nota: diferentes respostas são possíveis; escolha apenas uma.*

Guia para resolução: o último estado resulta de duas escritas emitidas concorrentemente, uma iniciada por FE1, outra por FE2. Como ambos os FE consultam a tag atual do sistema concorrentemente, ambos observam a mesma tag inicial. Para ver um exemplo de resposta, consultar os slides da disciplina (situação de escritas concorrentes).

Grupo II [3 valores]

Num sistema de *bike-sharing*, existem diferentes servidores: desde os servidores que gerem cada estação de bicicletas aos servidores que mantêm as contas de utilizadores. Num destes sistemas, considere a seguinte operação levantarComReserva, implementada como uma transação distribuída:

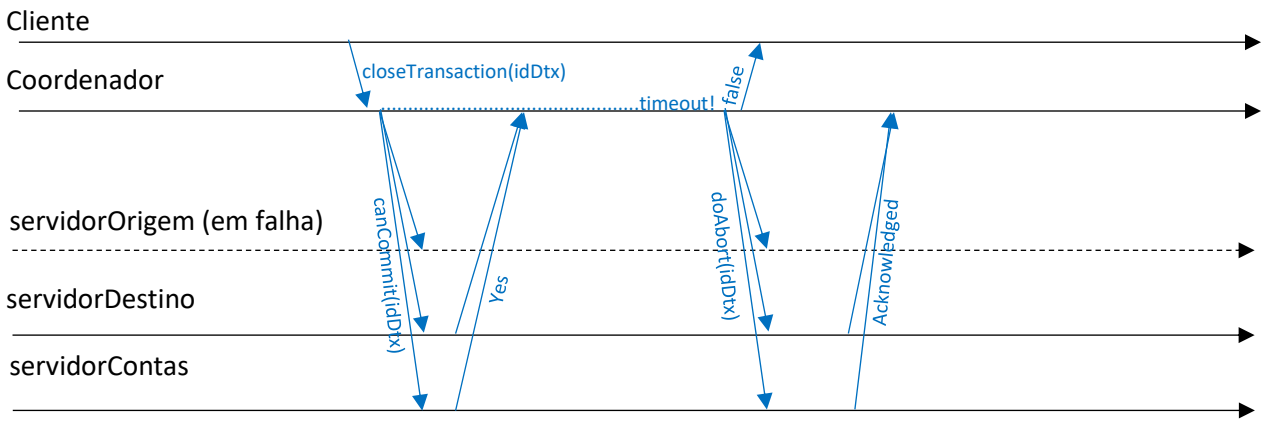
```
1 levantarComReserva (string utilizador, int estacaoOrigem, int estacaoDestino) {
2       servidorOrigem = lookup(estacaoOrigem);
3       servidorDestino = lookup(estacaoDestino);
4       servidorContas = lookup("utilizadores");

5       idDTx = coord.openTransaction();
6       try {
7               servidorOrigem.levantarBicicleta(utilizador, idDTx);
8               servidorDestino.reservarVaga(utilizador, idDTx);
9               servidorContas.debitar(utilizador, TAXA_RESERVA, idDTx);
10              coord.closeTransaction(idDTx);
11      } catch (Exception e) {
12              coord.abortTransaction(idDTx);
        }
}
```

1. [0,7v] Compare a solução acima com uma alternativa que não recorria a uma transação distribuída (linhas 5, 10, 12 não existiam). Descreva uma situação que pode ocorrer na versão não transacional e que não é possível na solução acima.

2. [0,7v] A operação `levantarComReserva` foi chamada sobre a conta do utilizador A, com sucesso. Em paralelo, assuma que existe um cliente concorrente que está regularmente a consultar o saldo dos utilizadores do sistema (lendo do mesmo servidor de contas, "utilizadores"). A partir de que instante é que este cliente concorrente observará o novo saldo do utilizador A?

3. [1,0v] Se, ao chegar à linha 10, um dos participantes estiver em falha demorada, como pode o coordenador lidar com essa situação? Responda preenchendo o diagrama de mensagens abaixo.



4. [0,6v] Em que momento pode o Coordenador limpar dos seus registos a transação? Justifique.

A partir do momento em que receba, por parte de *todos* os participantes (incluindo aqueles em falha temporária), a confirmação de que receberam e aplicaram a decisão de terminação da transação distribuída (*doCommit* ou *doAbort*).

Grupo III [10 valores]

1) Pretende-se **proteger informação confidencial** guardada num servidor ligado à Internet. Caso seja obtida por um atacante, esta informação vale garantidamente **EUR 5 000**.

   a) [0,7v] Assuma o papel da **defesa** e considere os seguintes custos de proteção:
   - verificador de senhas EUR 100,
   - base de dados cifrada EUR 500,
   - *firewall* para filtrar/limitar pedidos suspeitos EUR 1 000,
   - sistema operativo com segurança reforçada EUR 2 000,
   - servidores geo-replicados para resistir a muitos pedidos (*denial-of-service*) EUR 5 000.

   Considerando as opções anteriores, faça uma proposta para proteger o sistema.
   Indique qual o **critério** que utilizou para a sua proposta.

   b) [0,7v] Assuma agora o papel do **atacante** com motivos apenas de proveito económico.
   Considere os seguintes custos de ataque:
   - tentar senhas em dicionário EUR 500,
   - encontrar vulnerabilidade no sistema operativo EUR 1 000,
   - enviar muitos pedidos para negação-de-serviço (*denial-of-service*) EUR 3 000,
   - tentar senhas exaustivamente (força-bruta) EUR 5 000,
   - comprar vulnerabilidade no sistema operativo no mercado negro EUR 10 000.

   Qual é a combinação de ataques que propõe? Justifique.

2) [0,9v] Considera a **criptografia** uma política ou um mecanismo de segurança? Justifique a sua resposta indicando a diferença entre os dois conceitos.

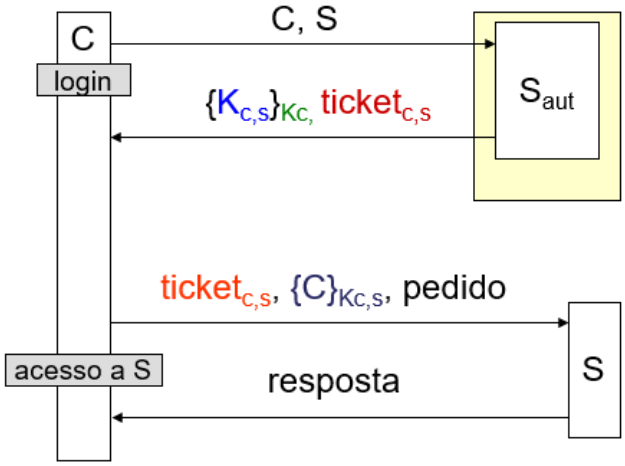
3) [0,7v] Considere uma cifra simétrica por blocos **sem** realimentação, como o AES ECB.
Qual é a vulnerabilidade?
   A. Caso o IV seja diferente, os blocos B1 e B2 são cifrados diferentemente mesmo que o seu conteúdo em claro seja igual.
   B. O criptograma produzido para um bloco B2 é sempre igual ao criptograma do bloco B1 anterior.
   C. O criptograma produzido para um bloco B1 é igual ao criptograma produzido para um bloco B2 se o conteúdo dos dois blocos for igual.
   D. Não há vulnerabilidade pois apenas é necessário um bloco de cifra de tamanho variável.

   C

4) [0,7v] Compare a segurança prática do AES-128 com o AES-256, assumindo que a chave é corretamente gerada e armazenada.
   A. Ambos os AES são menos seguros que o DES.
   B. Ambos oferecem o mesmo nível de segurança e apenas o tamanho do bloco de dados é diferente.
   C. O AES-256 oferece o dobro da segurança do AES-128.
   D. O AES-256 oferece muito mais do que o dobro da segurança do AES-128.

   D

5) Pretende-se proteger um sistema com diversos clientes e servidores com uma implementação alternativa do protocolo Kerberos simplificado apresentado nas aulas teóricas. Este protocolo alternativo chama-se ***QuerBerros*** e está ilustrado na figura abaixo.



   a) [0,7v] Qual é a distribuição de chaves que existe no sistema **antes** e **depois** de se executar o protocolo?

Entidade	Chaves conhecidas à partida	Chaves conhecidas no fim
C		
S		
Saut		

b) [0,9v] Qual é o conteúdo do $ticket_{c,s}$ apresentado na figura? Explicite para que serve cada elemento.

{ C, S, T1, T2, Kc,s } Ks

C – identificador de cliente
S – identificador do servidor
T1, T2 – intervalo de validade do ticket
Kc,s – chave de sessão para comunicação entre cliente e servidor
O ticket é cifrado com a chave do servidor Ks

c) [0,6v] O protocolo QuerBerros removeu duas proteções existentes no Kerberos. Identifique o que foi removido.

1ª proteção removida: *nonce* na primeira troca de mensagens

2ª proteção removida: *request time* na segunda troca de mensagens

d) [0,9v] Descreva um ataque que seja possível realizar devido à remoção de uma das proteções.

Nome do ataque: *replay attack*

Objetivo do atacante (o que tem a ganhar): repetir um pedido para o servidor

(por exemplo, assumindo que o pedido fazia um depósito bancário, repetir o depósito)

Descrição: o atacante deve capturar um pedido que o cliente envia para o servidor.

Mais tarde, mas ainda dentro da validade do ticket, o atacante reenvia o pedido e restantes

elementos. O servidor, como não tem forma de verificar a frescura, repete a execução

(não consegue distinguir um pedido novo de um pedido repetido).

(Outras respostas corretas são possíveis)

6) Pretende-se agora criar um sistema de autenticação para clientes e servidores de Web Services usando criptografia RSA.

a) [0,7v] Com a cifra assimétrica RSA é possível:

A. Cifrar com a chave pública, decifrar com a chave privada.
B. Cifrar com a chave privada, decifrar com a chave pública.
C. Combinar com cifra simétrica e fazer cifra híbrida.
D. Todas as anteriores.

D

b) [0,7v] Um certificado digital de chave pública confiável **não** pode conter:

A. O par de chaves da entidade certificada.
B. A chave pública da entidade.
C. A chave pública e o nome da entidade certificada.
D. A chave pública, o nome da entidade certificada e a assinatura de uma autoridade de certificação.

A

c) [0,8v] Pretende-se proteger uma mensagem SOAP para garantir a sua confidencialidade. Descreva abaixo qual o conteúdo da mensagem protegida. Proponha uma solução eficiente, tendo em conta que a dimensão da mensagem a enviar é muito superior ao tamanho de um bloco de cifra RSA.

Assuma que o cliente apenas conhece a sua chave privada K_{priv_C} e a chave pública do servidor K_{pub_S} e vice-versa.

```
<soap:envelope>
    <soap:header>
        <messageKey> { K, timestamp atual }Kpub_s </messageKey>
    </soap:header>



    <soap:body>
        <cipherBody> { body }K </cipherBody>
    </soap:body>
</soap:envelope>
```

(outras soluções também foram aceites como corretas)

d) [1,0v] Proponha agora *Handlers* que permitam, de forma **modular**, garantir as seguintes combinações de proteção: *apenas integridade*, *apenas confidencialidade, integridade com confidencialidade*.

SOAP Handlers propostos (o *Handler Log* é apresentado como exemplo):

H_{log} – imprime toda a mensagem SOAP (cabeçalho e corpo) no terminal do cliente
Hsign – cria uma assinatura digital da mensagem (resumo cifrado com chave privada emissor)
Hcipher – cifra a mensagem com uma chave simétrica - ver resposta da alínea c)

Cadeia cliente para garantir apenas integridade:

1º H_{log}, 2º Hsign

Cadeia cliente para garantir apenas confidencialidade:

1º H_{log}, 2º Hcipher

Cadeia cliente para garantir integridade e confidencialidade:

1º H_{log}, 2º Hsign, 3º Hcipher