Sistemas Distribuídos - 2ªChamada - Versão A

IST - LEIC-A/ LEIC-T/ LETI - 2023-2024
4 de julho de 2024

- A classificação máxima é de 20 pontos.
- A classificação mínima para aprovação é de 8 valores.
- Todas as respostas devem se dadas na "Folhas de Respostas".
- Identifique com o seu número e nome *todas* as folhas de resposta.
- Não pode sair da sala durante a primeira hora do exame.
- A utlização de telemóveis ou de equipamentos informáticos durante o exame é proibida.
- Nas respostas erradas às perguntas de escolha múltipla é descontada a cotação da pergunta dividida pelo número de alternativas.
- O exame tem a duração de 2 horas.

Chamada a Procedimentos Remotos

Questão 1 (*0.5 valor*) Considere os seguintes mecanismos que podem ser usados na concretização de um sistema de chamadas a procedimentos remotos: i) retransmissão de pedidos; ii) detecção de duplicados. Diga quais destes mecanismos são necessários para oferecer as seguintes semânticas:

- No máximo uma vez
- Pelo menos uma vez
- Exactamente uma vez

Questão 2 (*0.5 valor*) Considere um serviço de geração de números pseudo-aleatórios que oferece a operação remota *getRandom*, que produz uma sequência determinística de valores distintos Esta operação não recebe argumentos. A cada pedido, o servidor retorna o próximo valor nessa sequência, de nome *value*, e incrementa um índice (que mantém no estado do servidor)

Pretende-se desenvolver este serviço em gRPC. Complete os elementos na interface remota no excerto seguinte do ficheiro .proto.

```
syntax = "proto3";
package sd;

message m1 {

}

message m2 {

}

Randomiser {
    rpc getRandom(             ) returns (             ) ;
}
```

Questão 3 (*1 valor*) Complemente a interface remota com uma nova operação, chamada *getRandomIdempotent*, que permita uma implementação idempotente da operação original.

Sincronização de Relógios

Questão 4 (*1 valor*) Usando o algoritmo de Cristian, um cliente tenta sincronizar o seu relógio com um servidor de tempo. O cliente não conhece o tempo mínimo de propagação neste rede.

À primeira tentativa, o cliente recebe do servidor a leitura 1000, tendo o cliente observado que passaram 12 ms de RTT (round-trip time) entre o envio do seu pedido e a receção da resposta do servidor. Qual o novo valor que o cliente aplica ao seu relógio? Justifique apresentando os cálculos.

Questão 5 (*1 valor*) Assuma agora que o cliente sabe que o tempo mínimo de propagação (entre cliente e servidor) é 1 ms.

O cliente deseja sincronizar o seu relógio local com uma precisão de ± 3 ms (ou inferior). Para tal, o cliente irá repetir pedidos de sincronização até alcançar a precisão pretendida.

Hipoteticamente, considere que, caso o cliente repita pedidos ao servidor, o cliente observará a seguinte sequência de RTT (round-trip time): 12 ms, 8 ms, 15 ms, 5 ms.

Ao fim de quantos pedidos é que o cliente obtém a precisão desejada e dá a sincronização por terminada? Justifique apresentando os cálculos.

Relógios Lógicos

Considere a execução ilustrada na Figura 1.

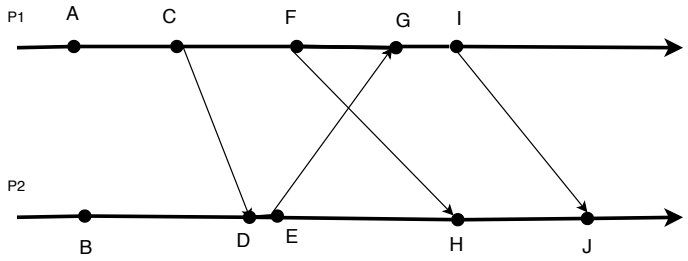


Figura 1: Execução distribuída

Questão 6 (*1 valor*) Considere que usa relógios lógicos de Lamport para marcar todos os eventos (isto é, tanto os eventos de emissão como os eventos de recepção de mensagens). Assuma que os eventos A e B foram marcados com os seguintes valores de tempo lógico: $l(A) = 10$ e $l(B) = 10$. Qual é o valor do relógio atribuído ao evento I?

Questão 7 (*1 valor*) Considere que usa relógios vectoriais para marcar todos os eventos (isto é, tanto os eventos de emissão como os eventos de recepção de mensagens). Assuma que os eventos A e B foram marcados com os seguintes relógios vectoriais: $vector(A) = (1, 0)$ e $vector(B) = (0, 1)$. Qual é o valor do relógio vectorial atribuído ao evento I?

Gossip - Lazy Replication

Considere o sistema replicado conhecido por "Lazy Replication"ou "Gossip", no qual as operações são propagadas "nos bastidores" por propagação epidémica. Considere um sistema com 3 réplicas, em que o estado de cada réplica é capturado por dois relógios vectoriais, *valueTS* e *replicaTS*. Assuma que todos os pedidos recebidos por uma réplica foram já aplicados nessa réplica, pelo que *valueTS* possui o mesmo valor que *replicaTS*, que passamos simplesmente a designar por S. Considere que num dado instante, os servidores encontram-se no seguinte estado: $S_1 =(1, 2, 5)$, $S_2 = (1, 7, 5)$ and $S_3 = (1, 4, 6)$.

Questão 8 (*0.5 valor*) Considere que um cliente com *prev*=(1, 4, 6) envia um pedido de leitura ao servidor S_1. Indique qual a afirmação verdadeira.

A) S_1 responde de imediato ao cliente.

B) S_1 devolve erro pois as réplicas só processam pedidos de escrita.

C) S_1 não responde de imediato mas, após S_1 se sincronizar com um dos outros servidores, S_1 já consegue responder ao cliente.

D) O cliente precisa contactar pelo menos mais um servidor para obter um quórum de respostas.

Questão 9 (*0.5 valor*) Até este instante, quantas escritas já foram aceites no sistema replicado? Assuma que um cliente, quando pretende invocar uma escrita, a envia exclusivamente a um dos servidores.

Exclusão Mútua

Considere o algoritmo descentralizado de Maekawa para exclusão mútua. Considere um sistema com 3 processos, p_1, p_2 e p_3. Assuma os seguintes conjuntos de votos associados a cada processo: $V_1 = \{p_1, p_2\}$, $V_2 = \{p_2, p_3\}$, $V_3 = \{p_3, p_1\}$.

Neste algoritmo o estado de cada processo é capturado por 3 variáveis, nomeadamente:

- *state*, que pode ter os valores RELEASED, WANTED, ou HELD.
- *voted*, de tipo booleano.
- *pending*, uma fila de pedidos.

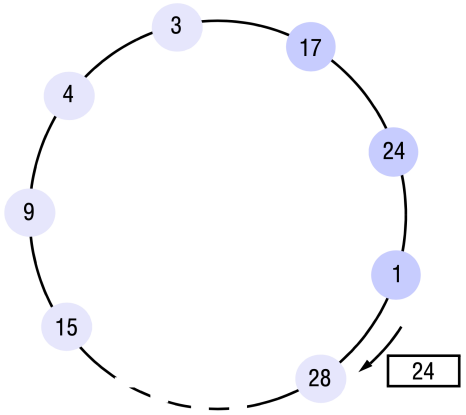
Considere que, num dado instante, o estado do sistema é o seguinte:

processo	p_1	p_2	p_3
state	HELD	WANTED	RELEASED
voted	TRUE	TRUE	TRUE
pending	$\emptyset$	p_2	$\emptyset$

Questão 10 (*1 valor*) Assuma que, a partir deste estado, o processo p_1 liberta o recurso. Descreva o estado do sistema após todas as mensagens do algoritmo terem sido trocadas.

Eleição de Líder

Considere o algoritmo de eleição de líder num anel que usa mensagens do tipo "election" e "elected". Considere o sistema ilustrado na figura seguinte:



Questão 11 (*0.5 valor*) Qual a mensagem que o nó 28 transmite depois de receber a mensagem "election=24"?

Questão 12 (*0.5 valor*) Qual a mensagem que o nó 28 transmite depois de receber a mensagem "election=28"?

Salvaguardas

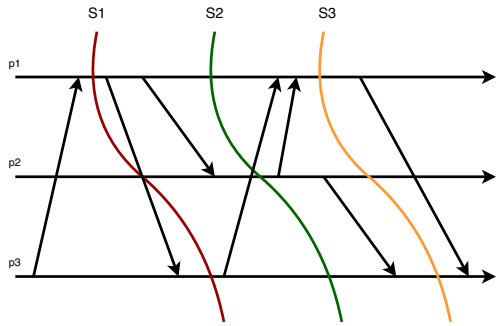


Figura 2: Cortes $S1$, $S2$ e $S3$

Questão 13 (*1 valor*) Considere a execução ilustrada na Figura 2. Para cada um dos cortes $S1$, $S2$ e $S3$, diga se o esse corte poderia ter sido obtido pelo algoritmo de Chandy-Lamport ou não.

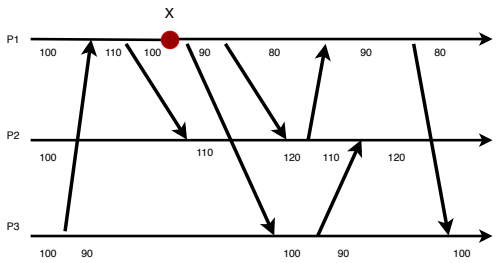


Figura 3: Salvaguarda distribuída

Questão 14 (*1 valor*) Considere a execução ilustrada na Figura 3, onde cada processo possui n tokens (inicialmente 100) e cada mensagem transfere 10 tokens entre dois processos. Considere que o processo p_3 inicia uma salvaguarda no instante X, executando o algoritmo de Chandy-Lamport. Qual vai ser o estado capturado pelo algoritmo?

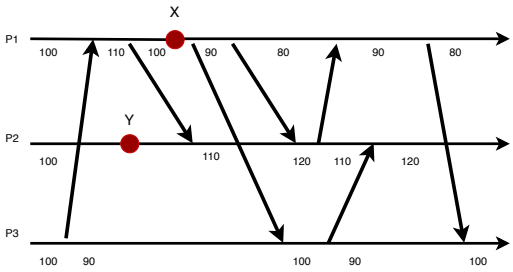


Figura 4: Salvaguarda distribuída (início concorrente)

Questão 15 (*1 valor*) Considere agora a execução ilustrada na Figura 4, na qual o processo p_2 incia no momento Y, e de forma concorrente com p_3 uma salvaguarda. Qual vai ser o estado capturado pelo algoritmo?

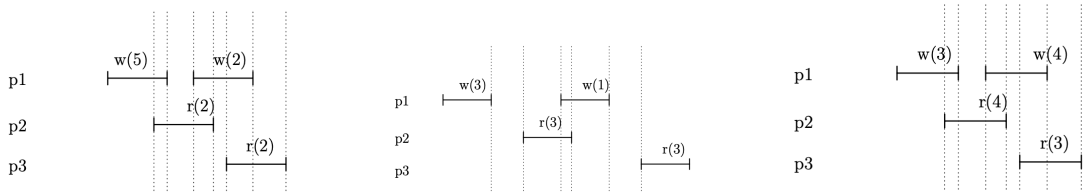


Figura 5: Execuções de registos

Registos

Questão 16 (*1 valor*) Considere as execuções de acesso a um registo ilustradas na Figura 5. Apenas uma das execuções é válida para registos atómicos, outra só válida para registos regulares (mas não atómicos) e outra não é válida nem em registos atómicos nem em registos regulares (poderá ser *safe* ou *unsafe*). Diga a que categoria corresponde cada execução.

Questão 17 (*1 valor*) Considere o algoritmo ABD para concretizar registos atómicos distribuídos. Neste algoritmo, para realizar uma leitura...

A) Basta ler a réplica local

B) Basta ler de uma maioria e escolher o valor mais recente

C) É preciso ler de uma maioria, escolher o valor mais recente, e escrever esse valor numa maioria

D) É preciso ler de uma maioria, escolher o valor mais recente, e escrever esse valor na réplica local

Ordem Total

Considere o algoritmo para estabelecer uma ordem total inventado pelo Dale Skeen (também designado por "acordo colectivo"). Neste algoritmo, cada receptor mantém uma fila ordenada de mensagens, em que cada entrada na fila é um tuplo com o seguinte formato:

⟨ id_da_mensagem, emissor, número_de_sequência, estado (**T**entativo ou **F**inal) ⟩.

Considere um sistema com três réplicas e vários clientes que enviam mensagens para estas réplicas usando o algoritmo de acordo colectivo. Considere que, num dado instante, o estado das réplicas é o seguinte:

réplica		
r_1	r_2	r_3
entregues		
	$\langle A, c_1, 1, F\rangle$	$\langle A, c_1, 1, F\rangle$
pendentes		
$\langle A, c_1, 1, T\rangle$	$\langle D, c_2, 2, T\rangle$	$\langle D, c_4, 2, T\rangle$
$\langle C, c_2, 2, T\rangle$	$\langle B, c_4, 3, T\rangle$	$\langle B, c_4, 3, T\rangle$
$\langle D, c_3, 3, T\rangle$	$\langle E, c_5, 4, T\rangle$	
$\langle B, c_4, 4, T\rangle$		

Questão 18 (*1 valor*) Qual será a próxima mensagem a ser entregue (isto é, imediatamente após A na ordem total)? Qual será o número final atribuído a esta mensagem? Justifique.

Consenso

Questão 19 (*1 valor*) Considere o problema do consenso. Cada processo propõe um valor, e todos devem acordar num único output que é:

A) Qualquer um dos valores propostos

B) O valor proposto pela maioria

C) Um vector com todos os valores propostos

D) A soma de todos os valores propostos

Questão 20 (*1 valor*) Num sistema distribuído com 4 processos, executou-se o algoritmo "floodset consensus" (estudado nas aulas teóricas), em 3 rondas e usando a função "mínimo" para a escolha do valor. Assuma modelo síncrono. Os processos propuseram os seguintes valores: p_1 propôs 10, p_2 propôs 20, p_3 propôs 30 e p_4 propôs 40. No entanto, a meio da primeira ronda, p1 falhou e o seu valor só chegou a p2 mas não a p3 nem a p4. Antes de começar a segunda ronda, p_2 falha também. Qual foi o valor acordado por p_3 e p_4?

Transacções Distribuídas

Questão 21 (*1 valor*) Considere um participante no protocolo de confirmação atómica em duas fases (two-phase commit). O coordenador recebe OK de todos os participantes excepto de um participante que não envia nenhuma resposta. Neste caso, ocoordenador pode abortar a transacção? Justifique.

Segurança e Canais Seguros

Questão 22 (*1 valor*) Assuma que os participantes A e B possuem um par de chaves assiméricas, $\langle A^-, A^+\rangle$ e $\langle B^-, B^+\rangle$ respectivamente, em que as chaves públicas de ambos foram corretamente distribuídas, e que partilham um função de *hash* criptográfica e funções de cifra simétrica e assimétrica. Assuma também que A pode gerar uma chave simétrica K_{ab}.

Considere que A quer enviar uma mensagem m secreta para outro participante B. Para tal, A envia o seguinte: $\langle \{m\}_{K_1}, \{K_2\}_{K_3}\rangle$

Indique a que chaves correspondem as variáveis K_1, K_2, K_3.

Questão 23 (*1 valor*) Assuma que A e B trocavam para este outro protocolo:

$$\langle \{m, C(A)\}_{K_1}, \{K_2\}_{K_3}, \{hash(\{m, C(A)\})\}_{A^-}\rangle$$

Assuma que A e B mantêm relógios lógicos de Lamport, C(A) e C(B). Que propriedades são asseguradas com este protocolo que não eram com o protocolo anterior?

A) confidencialidade

B) autentiticade

C) não repúdio

D) integridade

E) frescura

F) disponibilidade

Folha de Respostas (1/4): não dobrar esta folha

IST ID:	**Nome:**	**Versão:**

§

		Retransmissão de pedidos	Detecção de duplicados
Questão 1	Pelo menos uma vez		
	No máximo uma vez		
	Exactamente uma vez		

Questão 2

```
message m1                          {



}

message m2                          {


}

service randomizer {
     rpc getRandom         (                ) returns (         )
}
```

Questão 3

```
message                             {



}

message                             {


}

service randomizer {
     ...
     rpc                 (                ) returns (                 )
}
```

§

Sincronização de relógios:	
Questão 4	
Questão 5	

Folha de Respostas (2/4): não dobrar esta folha

§

IST ID:	**Nome:**	**Versão:**

§

Relógios lógicos:		
Questão 6	evento I:	
Questão 7	evento I:	

§

Lazy Replication	
Questão 8	
Questão 9	

Mutual Exlusion

		p_1	p_2	p_3
Questão 10	*state*			
	voted			
	pending			

§

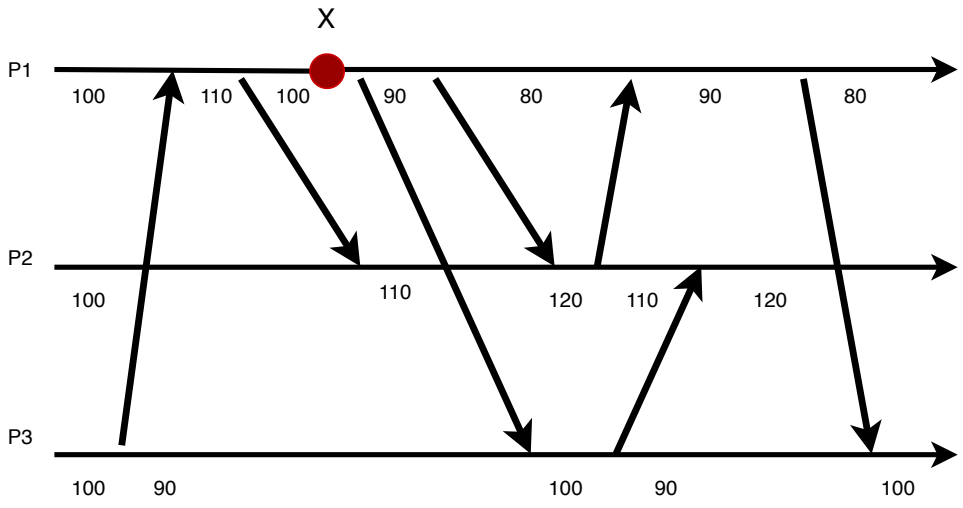
Eleição de líder:	
Questão 11	
Questão 12	

§

Cortes coerentes:			
Questão 13	S_1:	◯ sim	◯ não
	S_2:	◯ sim	◯ não
	S_3:	◯ sim	◯ não

Chandy-Lamport:

Questão 14	p_1:	p_2:	p_3:
	c_{11}: $\emptyset$	c_{12}:	c_{13}:
	c_{21}:	c_{22}: $\emptyset$	c_{23}:
	c_{31}:	c_{32}:	c_{33}: $\emptyset$
	(ilustre a execução na figura abaixo)		



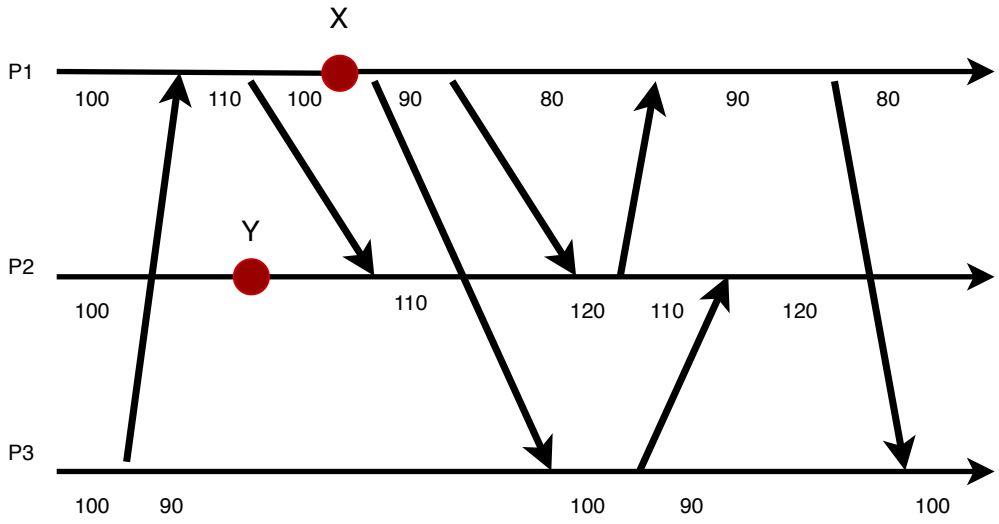
Folha de Respostas (3/4): não dobrar esta folha

IST ID:	**Nome:**	**Versão:**

§

Chandy-Lamport:

Questão 15	p_1:	p_2:	p_3:
	c_{11}: $\emptyset$	c_{12}:	c_{13}:
	c_{21}:	c_{22}: $\emptyset$	c_{23}:
	c_{31}:	c_{32}:	c_{33}: $\emptyset$
	(ilustre a execução na figura abaixo)		



§

Registos:

	Caso	Unsafe	Safe	Regular	Atómico
Questão 16	a)				
	b)				
	c)				
Questão 17					

§

Ordem Total:

Questão 18	Mensagem:
	Número final:
	justificação: .

Folha de Respostas (4/4): não dobrar esta folha

IST ID:	**Nome:**	**Versão:**

§

Consenso	
Questão 19	
Questão 20	

§

Transacções:	
Questão 21	◯ pode abortar ◯ não pode abortar justificação: .

§

Segurança:	
Questão 22	K_1? K_2? K_3?
Questão 23	☐ confidencialidade? ☐ autenticidade? ☐ não repúdio? ☐ integridade? ☐ frescura? ☐ disponibilidade?

Soluções (1/4)

IST ID:	**Nome:**	**Versão:** A	

§

		Retransmissão de pedidos	Detecção de duplicados
Questão 1	Pelo menos uma vez	✓	
	No máximo uma vez	(✓)	✓
	Exactamente uma vez	✓	✓

Questão 2

```
message m1 {
}

message m2 {
    float value = 1;
}

service randomizer {
     rpc getRandom (1) returns (m2)
}
```

Questão 3

```
message m3 {
    int index = 1;
}

message m4 {
    float value = 1;
}

service randomizer {
     ...
     rpc getRandomIndempotent(m3) returns (m4)
}
```

§

Sincronização de relógios:	
Questão 4	$1000 + \frac{12}{2} = 1006$
Questão 5	2^{o} pedido $\frac{12-2}{2} = 5 > 3$ $\frac{8-2}{2} = 3$

Soluções (2/4)

§

IST ID:	**Nome:**	**Versão:** A	

§

Relógios lógicos:		
Questão 6	evento I:	15
Questão 7	evento I:	$[5, 3]$

§

Lazy Replication	
Questão 8	C
Questão 9	14

Mutual Exlusion

		p_1	p_2	p_3
Questão 10	*state*	RELEASED	HELD	RELEASED
	voted	FALSE	TRUE	TRUE
	pending	$\emptyset$	$\emptyset$	$\emptyset$

§

Eleição de líder:	
Questão 11	election=28
Questão 12	elected=28

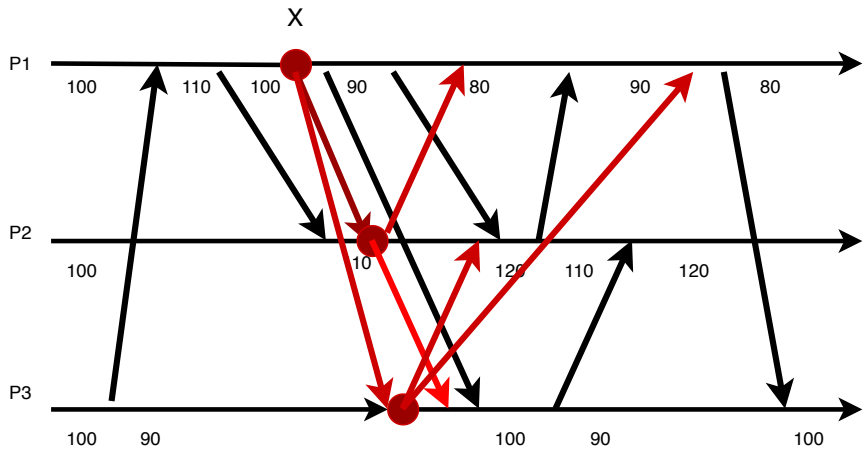
§

Cortes coerentes:

Questão 13	S_1:	○ sim	● não
	S_2:	● sim	○ não
	S_3:	● sim	○ não

Chandy-Lamport:

Questão 14	p_1: 100	p_2: 110	p_3: 90
	c_{11}: $\emptyset$	c_{12}: $\emptyset$	c_{13}: $\emptyset$
	c_{21}: $\emptyset$	c_{22}: $\emptyset$	c_{23}: $\emptyset$
	c_{31}: $\emptyset$	c_{32}: $\emptyset$	c_{33}: $\emptyset$
	(ilustre a execução na figura abaixo)		



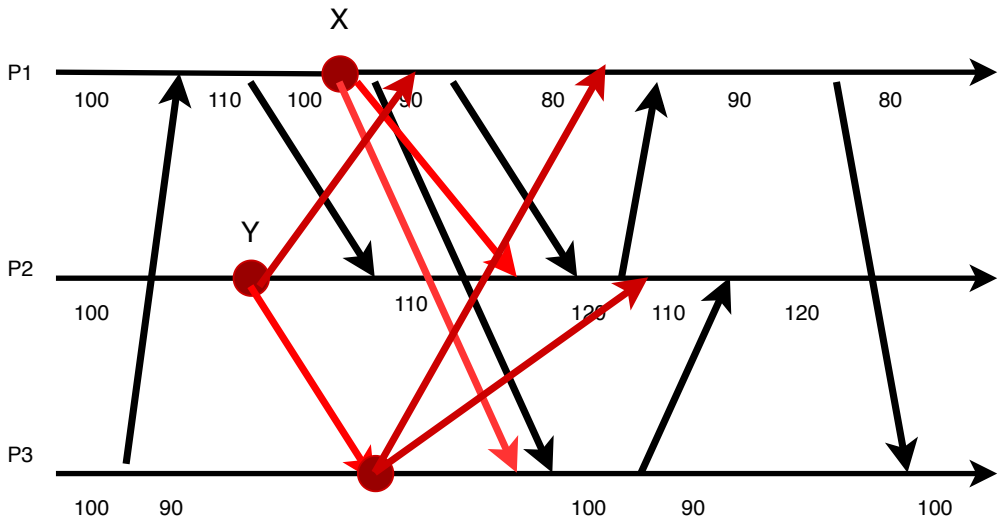
Soluções (3/4)

IST ID:	**Nome:**	**Versão:** A	

§

Chandy-Lamport:

Questão 15	p_1: 100	p_2: 100	p_3: 90
	c_{11}: $\emptyset$	c_{12}: 10	c_{13}: $\emptyset$
	c_{21}: $\emptyset$	c_{22}: $\emptyset$	c_{23}: $\emptyset$
	c_{31}: $\emptyset$	c_{32}: $\emptyset$	c_{33}: $\emptyset$
	(ilustre a execução na figura abaixo)		



§

Registos:

	Caso	Unsafe	Safe	Regular	Atómico
Questão 16	a)				✓
	b)	✓			
	c)			✓	

Questão 17	C

§

Ordem Total:

Questão 18	Mensagem: D Número final: 3 justificação: $B = 4$ $C > 4$ $D = 3$ $E > 4$ $F, G, H, \ldots > 4$

Soluções(4/4)

IST ID:	**Nome:**	**Versão:** A	

§

Consenso	
Questão 19	A
Questão 20	20

§

Transacções:

Questão 21	● pode abortar ○ não pode abortar justificação: . Nenhum participante sabe ainda a decisão

§

Segurança:

Questão 22	$K_1 = K_{ab}$ $K_2 = K_{ab}$ $K_3 = B^+$
Questão 23	☐ confidencialidade? ☒ autenticidade ☒ não repúdio ☒ integridade ☒ frescura ☐ disponibilidade