

GABARITO CPD

1) a.

	1	2	4	8	12	16
$T(p)$	1000	550	280	140	100	90
$S(p)$	1	1,82	3,57	7,14	10	11,11
$E(p)$	1	0,91	0,89	0,83	0,83	0,63

b. 8 processadores, porque a eficiência ainda se mantém alta mesmo com o aumento dos processadores, ou 12 processadores, quando a eficiência ainda não caiu

2) a. $S(8) = \frac{1}{\frac{0,03 + 1 - 0,03}{8}} = 6,61$

b. $S(p) = \frac{1}{\frac{0,03 + (1 - 0,03)}{p}} = \frac{1}{0,03} = 33,3$

c. $S(8) = 8 - 0,03 \cdot (8 - 1) = 7,79$

d. Amdahl \rightarrow considera tamanho fixo do problema

Gustafson \rightarrow considera aumentar o tamanho do problema

Gabarito comentado da prova

Questão 3

A) “Aumenta a disponibilidade e reduz a latência.”

✗ Falso. Consistência forte faz o oposto: reduz disponibilidade e aumenta latência, porque exige coordenação entre réplicas antes de confirmar operações.

B) “Mantém as réplicas sempre sincronizadas, à custa de maior latência e menor disponibilidade.”

✓ Verdadeiro. Esse é exatamente o impacto da consistência forte:

- exige que todas (ou um conjunto de quorum) réplicas concordem antes de responder;
 - isso aumenta a latência, pois há comunicação extra;
 - também reduz a disponibilidade, pois operações podem falhar se nem todas as réplicas necessárias estiverem acessíveis.
-

C) “Reduz o número de mensagens trocadas entre réplicas.”

✗ Falso. O número de mensagens aumenta — coordenação é necessária para manter os dados sincronizados.

D) “Garante desempenho máximo em cenários de falha de rede.”

✗ Falso. Com consistência forte, falhas de rede frequentemente impedem operações (CAP: consistência → menor disponibilidade).

E) “Opera de forma idêntica à consistência eventual.”

✗ Falso. São modelos opostos:

- consistência eventual aceita divergência temporária;
 - consistência forte exige sincronia imediata.
-

✓ Alternativa correta: B

Questão 4

A) “As falhas sempre ocorrem simultaneamente em todos os nós.”

✗ Falso. Falhas em sistemas distribuídos são independentes e muitas vezes imprevisíveis. Não é por ocorrerem simultaneamente que a transparência é difícil — isso quase nunca acontece.

B) “Depende apenas de mecanismos de segurança.”

✗ Falso. Transparência de falhas envolve muito mais que segurança:

- detecção de falhas
 - recuperação
 - replicação
 - máscaras de falhas
 - consistência
- Segurança é apenas um aspecto.

C) “Requer um único ponto central de controle para reiniciar processos.”

✗ Falso. Pelo contrário: um ponto central seria um *single point of failure*, piorando a tolerância a falhas. Sistemas realmente tolerantes precisam ser descentralizados.

D) “É necessário detectar, mascarar e recuperar falhas em um ambiente onde a comunicação pode ser não confiável.”

✓ Verdadeiro. Este é o motivo real e fundamental.

Sistemas distribuídos enfrentam:

- comunicação assíncrona
- latência variável
- perda e duplicação de mensagens
- processos que podem falhar silenciosamente

Implementar transparência significa fazer o sistema parecer livre de falhas, mesmo com esse ambiente adverso — o que é intrinsecamente difícil.

✓ Alternativa correta: D

Questão 5

a) (F)

Falso. A arquitetura cliente-servidor não é totalmente descentralizada — ela depende de servidores dedicados, que fornecem serviços aos clientes.

b) (V)

Verdadeiro. Em N-Tier, separar apresentação, lógica e dados melhora:

- manutenção,
 - escalabilidade (especialmente horizontal),
 - reutilização e modularidade.
-

c) (V)

Verdadeiro. SOA organiza aplicações como serviços independentes, com comunicação via interfaces bem definidas, geralmente padronizadas (SOAP, WSDL, REST em alguns casos modernos).

d) (V)

Verdadeiro. Microserviços seguem muitos princípios da SOA, porém com:

- serviços menores,
- maior independência,
- maior automação e escalabilidade.

Por isso são vistos como uma evolução prática da SOA.

e) (F)

Falso. Arquiteturas orientadas a eventos são tipicamente assíncronas.

Em pub/sub, por exemplo, o emissor não espera resposta imediata.

f) (V)

Verdadeiro. Em P2P, todos os nós podem atuar como cliente e servidor, compartilhando recursos e responsabilidades.

g) (V)

Verdadeiro. Em uma DHT (Distributed Hash Table), chaves são distribuídas entre nós por meio de uma função de hash consistente, o que tende a garantir balanceamento de carga.

h) (V)

Verdadeiro. Chord usa a Finger Table para permitir buscas em $O(\log N)$, reduzindo drasticamente o número de saltos.

i) (V)

Verdadeiro. Chord funciona, mas alto churn (entrada/saída contínua de nós) dificulta a manutenção das suas tabelas de encaminhamento, exigindo estabilização frequente.

j) (F)

Falso. Escalabilidade não é apenas escalar verticalmente.

Escalabilidade inclui escala horizontal, distribuída, adicionando novos nós — fundamental em sistemas distribuídos.

Questão 6

a) “Comunicação baseada em eventos permite sistemas reativos.”

☒ Verdadeiro. Arquiteturas orientadas a eventos permitem que o sistema reaja automaticamente quando um evento ocorre, o que é exatamente o comportamento reativo.

b) “A comunicação é obrigatoriamente síncrona.”

☒ Falso. O modelo orientado a eventos é principalmente assíncrono.

O emissor publica o evento e não espera resposta imediata.

c) “Um evento pode disparar múltiplos serviços.”

☒ Verdadeiro. Em pub/sub, vários consumidores podem receber e reagir ao mesmo evento.

d) “Esse estilo favorece desacoplamento.”

☒ Verdadeiro. O produtor não precisa conhecer os consumidores — só publica o evento.

Isso reduz dependências diretas e aumenta modularidade.

✓ Alternativa correta: B) Apenas a, c, d

Questão 7

1. “A consistência forte garante que todas as leituras retornem o valor mais recente, mesmo em presença de partições de rede.”

☒ Falso. No modelo CAP, não é possível garantir consistência forte e tolerância a partições ao mesmo tempo.

Em caso de partição de rede, um sistema com consistência forte normalmente bloqueia operações para evitar leituras desatualizadas — ou seja, ele não garante operação normal.

2. “A consistência eventual admite leituras de dados desatualizados temporariamente.”

✓ Verdadeiro. Esse é exatamente o conceito: o sistema eventualmente converge, mas leituras podem retornar valores antigos até a sincronização ocorrer.

3. "No modelo de leitura monotônica, um cliente pode ler valores mais antigos do que já havia lido antes."

✗ Falso. Leitura monotônica garante que um cliente nunca leia uma versão mais antiga que aquela que já observou.

4. "O modelo de 'read-your-writes' garante que o cliente sempre veja o efeito de suas próprias escritas subsequentes."

✓ Verdadeiro. Se o cliente escreve algo, qualquer leitura posterior feita por ele deve refletir sua escrita.

5. "A consistência causal é mais fraca que a consistência eventual."

✗ Falso. A consistência causal é mais forte.

Ela impõe ordenação entre operações que possuem relação de causalidade.

A consistência eventual não impõe nenhuma ordem.

✓ Alternativa correta:

C) F – V – F – V – F

Questão 8

A) "Confiabilidade mede tempo médio entre falhas; disponibilidade mede se o sistema está acessível; tolerância a falhas é a capacidade de se recuperar automaticamente."

✓ Correta. Os três conceitos são diferenciados da forma clássica:

Confiabilidade: probabilidade de o sistema operar sem falhas durante um intervalo → normalmente expressa via MTBF (Mean Time Between Failures).

Disponibilidade: probabilidade de o sistema estar acessível quando solicitado → relacionada a MTBF e MTTR.

Tolerância a falhas: capacidade de continuar funcionando mesmo quando ocorrem falhas, usando redundância, replicação, etc.

B) "Confiabilidade e disponibilidade são sinônimas."

✗ Falso. Um sistema pode ser confiável (falha pouco) mas indisponível quando falha (se demora muito para recuperar).

C) "Um sistema pode ser altamente confiável mesmo sendo frequentemente indisponível."

✗ Falso nos termos usuais. Se é “frequentemente indisponível”, então tem baixa disponibilidade — mas também baixa confiabilidade, já que MTBF tende a ser baixo ou MTTR alto.

A frase confunde métricas e não está correta.

D) “Tolerância a falhas reduz a confiabilidade de um sistema.”

✗ Falso. Normalmente ocorre o oposto: tolerância a falhas aumenta confiabilidade e disponibilidade, pois o sistema continua operando mesmo quando algo falha.

E) “Confiabilidade e tolerância a falhas são conceitos equivalentes.”

✗ Falso. Tolerância a falhas é um mecanismo; confiabilidade é uma propriedade resultante. Não são equivalentes.

✓ Alternativa correta: A

Questão 9

A) O multicast atômico ignora falhas de processo. — Falsa

O multicast atômico não ignora falhas. Pelo contrário, sua definição exige que, mesmo com falhas, todas as réplicas entreguem *exatamente o mesmo conjunto e ordem* de mensagens (ou nenhuma entregue).

B) O multicast confiável garante ordenação total. — Falsa

O multicast *confiável* (reliable multicast) garante apenas entrega (sem perdas) e entrega única, mas não garante ordenação total.

Ordenação total é uma propriedade do multicast atômico (atomic/total order multicast).

C) No multicast confiável, todas as mensagens chegam, mas em ordem diferente; no multicast atômico, todas chegam exatamente na mesma ordem em todos os processos. — Verdadeira

Essa é exatamente a distinção clássica:

- Multicast confiável: garante que todos os processos recebem as mensagens, mas pode haver diferenças na ordem de entrega.
- Multicast atômico (total order): garante entrega confiável e que todos entregam na mesma ordem, preservando consistência entre réplicas.

Logo, é a alternativa correta.

D) O multicast confiável é sempre implementado sobre TCP. — Falsa

Não necessariamente. Multicast frequentemente utiliza UDP com protocolos adicionais no nível da aplicação. TCP, por definição, não é multicast.

E) O multicast atômico não garante entrega. — Falsa

O multicast atômico garante sim entrega confiável, além de ordenação total.

✓ Resposta correta: C