

TEORIA

Aquest examen consta de 40 qüestions. En cada cas només una de les respostes és correcta. Per a indicar la resposta n'hi ha prou amb emplenar la casella corresponent en la fulla de respostes adjunta. Totes les qüestions tenen el mateix valor. Si són correctes, aporten 0,25 punts a la nota obtinguda. Si són incorrectes descomptaran 1/5 del valor correcte, és a dir, - 0,05 punts. Convé pensar acuradament les respostes.

La durada d'aquesta part de l'examen és d'1 hora 40 minuts.

1. Els sistemes distribuïts...

A	...sempre estan formats per un conjunt d'agents concurrents que pot executar-se en un conjunt d'ordinadors interconnectats.
B	...proporcionen algun nivell de tolerància a fallades.
C	...permeten accedir a recursos compartits.
D	...poden utilitzar pas de missatges com el seu mecanisme de comunicació entre agents.
E	<p>Totes les anteriors.</p> <p>Tots els apartats anteriors són certs.</p> <p>Es necessiten múltiples agents. Tots ells han d'executar-se alhora, concurrentment. Es necessiten diversos ordinadors per a fer això. Això força al fet que hi haja una xarxa que els intercomunique.</p> <p>Els sistemes distribuïts han de proporcionar transparència de fallades (tolerància a fallades). D'una altra manera una fallada en un sol node podria provocar que el sistema deixara de funcionar.</p> <p>Els recursos poden ser compartits pels agents. D'aquesta manera es poden reduir els costos econòmics, incrementant l'ús efectiu dels recursos de maquinari adquirits.</p> <p>Poden utilitzar-se dos mecanismes d'intercomunicació: intercanvi de missatges i compartició de memòria.</p>
F	Cap de les anteriors.

2. Algunes de les àrees d'aplicació en computació distribuïda són...

A	<p>...les transparències d'ubicació, replicació, migració, persistència, transaccional, accés i fallades.</p> <p>La transparència és un objectiu en tot sistema distribuït. Per tant, tots els tipus de transparència que s'esmenten en aquest apartat son exemples d'objectius en un sistema distribuït, però cap d'ells és una àrea d'aplicació. Les àrees d'aplicació són tipus d'aplicacions informàtiques que tinguen especial sentit en el camp dels sistemes distribuïts.</p>
---	--

B	...WWW, xarxes de sensors, <i>Internet of Things</i> , computació cooperativa, <i>clusters</i> altament disponibles, computació en el núvol, etc. Cert. Aquestes van ser algunes de les àrees d'aplicació de la computació distribuïda presentades en el Tema 1.
C	...productor-consumidor amb buffer fitat, problema dels lectors-escriptors, problema dels cinc filòsofs, etc. Fals. Aquests són exemples de problemes clàssics de la programació concurrent.
D	...seccions crítiques, consens distribuït, difusió atòmica, pertinença a grup, consistència final, <i>commit</i> distribuït, transaccions niades, etc. Fals. Aquests són alguns conceptes generals, problemes o mecanismes que podem trobar en el camp de la computació distribuïda. Cap d'ells arriba a ser una aplicació (i molt menys una àrea d'aplicació).
E	Totes les anteriors.
F	Cap de les anteriors.

3. L'objectiu científicotècnic de la computació en el núvol és...

A	...dissenyar algorismes centralitzats. Fals. Els algorismes centralitzats solen introduir un únic punt de fallada (pel que no solen ser convenients per a garantir transparència de fallades) i “colls de botella” (és a dir, alguna cosa que introdueix una pèrdua de rendiment important). No pot recomanar-se el seu ús generalitzat per a dissenyar i implantar serveis distribuïts escalables. La computació en el núvol tracta de proporcionar serveis distribuïts escalables i adaptatius. Per tant, el disseny d'algorismes centralitzats hauria d'evitar-se.
B	...guanyar diners. Fals. Aquest no és un exemple d'objectiu científicotècnic.
C	...desplegar contenidors tolerants a fallades. Fals. Un contenidor no és una peça de programari que pugui ser desplegada, sinó l'element utilitzat per a desplegar components en ell. Aquesta frase no té sentit. A més, l'objectiu científicotècnic de la computació en el núvol no està restringit a l'etapa de desplegament.
D	...convertir la creació i explotació dels serveis de programari en una cosa més senzilla i eficaç. Aquest és precisament el seu objectiu principal. Així es va descriure en el Tema 1.
E	Totes les anteriors.
F	Cap de les anteriors.

4. En l'àrea de la computació en el núvol hi ha múltiples rols relacionats amb el cicle de vida d'un servei de programari. Aquests rols són...

A	Web, treballador i MV. Fals. Aquests són els tres tipus de rols o components en Windows Azure, però aquests rols estan relacionats amb com estructurar un servei en múltiples components. Aquesta qüestió no pregunta per aquesta classe de rols sinó per aquelles relacionades amb les activitats que es duen a terme durant les diferents etapes del cicle de vida d'un servei de programari.
---	--

B	Monitoratge, anàlisi, planificació, execució i coneixement (MAPE-K). Fals. Aquestes han sigut les cinc etapes o elements identificats en l'especificació original d'IBM per al cicle de vida en una aproximació de computació autònoma. No és l'objectiu d'aquesta qüestió.
C	Usuari, desenvolupador, administrador i proveïdor. Cert. Els rols que hem identificat en el Tema 1 són aquests quatre.
D	SaaS, PaaS i IaaS. Fals. Aquests no són rols relacionats amb activitats del cicle de vida sinó que són els tres models de servei que usualment es distingeixen en el camp de la computació en el núvol.
E	Totes les anteriors.
F	Cap de les anteriors.

5. Quina és la relació entre sistemes concurrents i sistemes distribuïts?

A	Tot sistema distribuït és també un sistema concurrent. Cert. Els sistemes distribuïts estan formats per agents que col·laboren per aconseguir un objectiu comú i que són executats en múltiples ordinadors independents que faciliten la imatge d'un sistema únic i coherent. Si hi ha múltiples agents que col·laboren entre si, llavors també tenim un sistema concurrent. Per tant, tenir un sistema distribuït també implica tenir un sistema concurrent.
B	Els sistemes concurrents no són sistemes distribuïts. Encara que alguns sistemes concurrents puguin estar compostos per agents que s'executen en un sol ordinador (i això implica un sistema que no és distribuït), un subconjunt dels sistemes concurrents està format per tots els sistemes distribuïts. Per això, aquesta afirmació no és certa en tots els casos. És a dir, hi ha sistemes concurrents que són sistemes distribuïts.
C	Els sistemes distribuïts no són sistemes concurrents. Fals. Tots els sistemes distribuïts són també sistemes concurrents.
D	Tot sistema concurrent és també un sistema distribuït. Fals. Hi ha múltiples exemples de sistemes concurrents que no necessiten executar-se en múltiples ordinadors. Per exemple, un procés Java que implante el problema dels cinc filòsofs seria un exemple d'aquest tipus. En poder-se implantar mitjançant un únic procés (per exemple, emprant un fil d'execució per a cada filòsof), no resulta possible parlar d'un sistema distribuït.
E	Totes les anteriors.
F	Cap de les anteriors.

6. En un sistema distribuït, els seus agents poden interactuar...

A	...utilitzant un mecanisme d'intercanvi de missatges. Cert. Tots els exemples que hem utilitzat en el seminari sobre ZeroMQ il·lustren aquest fet.
---	---

B	...seguint una aproximació client-servidor. Cert. Aquesta és una de les possibles aproximacions. Les aplicacions que utilitzen ROI, RPC o el patró de comunicació REQ/REP de ZeroMQ segueixen aquesta aproximació.
C	...seguint una aproximació <i>peer-to-peer</i> . Cert. Un exemple el proporcionen les aplicacions d'intercanvi de fitxers. Sota aquesta aproximació els agents no poden ser classificats com a clients ni com a servidors.
D	...compartint memòria. Cert. A més de l'intercanvi de missatges, els agents d'un sistema distribuït poden comunicar-se mitjançant memòria compartida. Tant un procés multi-fil com les diferents rèpliques d'un mateix component mantenen objectes de memòria que són utilitzats concurrentment per múltiples fils o processos.
E	Totes les anteriors.
F	Cap de les anteriors.

7. En un sistema distribuït...

A	...cada un dels seus agents té un estat privat i no interactua amb altres agents. Fals. Si no interactuaren amb la resta d'agents, els processos serien entitats independents i aïllades. En aquest cas seria un procés completament independent i no formaria part d'un sistema distribuït amb cert objectiu comú per a tots els seus agents.
B	...els agents poden tenir el seu propi estat, però col·laboren per a aconseguir un objectiu global. Cert. Cada agent pot tenir el seu propi estat, encara que l'estat no necessita ser "privat" (els seus elements poden estar compartits amb altres agents). A més, aquests agents han de col·laborar. Això és clau en els agents d'un sistema concurrent o distribuït. D'una altra manera, serien processos aïllats.
C	...els agents són independents i no comparteixen recursos. Fals. Ja s'ha explicat en els dos apartats anteriors.
D	...la concurrència és l'origen de molts problemes. Per tant, els sistemes distribuïts moderns no són concurrents. Fals. La concurrència és una propietat inherent als sistemes distribuïts; és a dir, els sistemes distribuïts han de ser concurrents.
E	Totes les anteriors.
F	Cap de les anteriors.

8. Les interaccions *peer-to-peer*...

A	...no s'utilitzen en sistemes distribuïts. Fals. És una de les aproximacions per a la interacció en sistemes distribuïts. L'altra és la interacció client-servidor.
---	--

B	...assumeixen que els agents estan interessats en alguna classe de recurs i quan un agent obté un d'aquests recursos, el distribueix a altres agents. Cert. Aquest és el nucli d'aquesta aproximació. Una vegada l'agent obté un recurs (encara que siga solament una part, com en els sistemes d'intercanvi de fitxers) pot oferir-ho a altres agents. Per tant, no hi ha cap distinció entre rol client i rol servidor. Tots els agents són capaços de comportar-se d'ambdues maneres: com a font i com a destinació de les transferències de recursos.
C	...distingeixen clarament entre agents clients i agents servidors. Fals. De manera general als agents d'un sistema peer-to-peer se'ls anomena "servents". Aquesta és la contracció de les paraules angleses <i>SERVER</i> i <i>CLIENT</i> .
D	...són un tipus d'interacció fortament centralitzada. Fals. Normalment els sistemes en els quals s'utilitza la interacció peer-to-peer estan descentralitzats.
E	Totes les anteriors.
F	Cap de les anteriors.

9. La WWW...

A	...és un exemple d'aplicació distribuïda que segueix un model d'interacció <i>peer-to-peer</i> . Fals. Aquesta àrea d'aplicacions distribuïdes sol utilitzar una interacció client-servidor.
B	...usa els navegadors web com un tipus específic d'agent servidor. Fals. Els navegadors web solen participar com a agents clients en la seua interacció amb altres agents.
C	...és un tipus d'àrea d'aplicació dels sistemes distribuïts on es transfereixen documents entre servidors i clients. Cert. En el cas més general, es transfereixen documents (per exemple, el contingut de les pàgines web estàtiques) entre servidors i clients.
D	...no utilitza un model d'interacció client-servidor. Fals. Sol utilitzar-se precisament aquest.
E	Totes les anteriors.
F	Cap de les anteriors.

10. Sobre els models de servei en la computació en el núvol:

A	IaaS: Proporciona aplicacions generals com el seu servei. Un exemple és Google Docs / Google Drive. Fals. Això és la definició i un exemple de model SaaS.
---	---

B	SaaS: Automatitza el desplegament i l'elasticitat de les aplicacions. Un exemple és Windows Azure. Fals. Això és la definició i un exemple de model PaaS.
C	PaaS: Proporciona una infraestructura virtual com el seu servei, on els components poden ser desplegats d'una manera no automatitzada. Exemple: Amazon EC2. Fals. Això és la definició i un exemple de model IaaS.
D	IaaS utilitza els serveis proporcionats per SaaS, que al seu torn utilitza els serveis del PaaS. Així es defineix una arquitectura de tres nivells. Fals. SaaS se situa sobre PaaS i aquest sobre IaaS.
E	Totes les anteriors.
F	Cap de les anteriors.

11. Les propietats que s'exigeixen a un sistema distribuït són...

A	Control centralitzat. No. El control centralitzat difícilment podrà escalar. L'escalabilitat és un objectiu dels sistemes distribuïts.
B	Actualitzacions diàries del programari. No. No sol ser necessari actualitzar amb tanta freqüència. Com en qualsevol altra àrea de la computació, les actualitzacions es necessiten per a corregir errors de programació o per a estendre la funcionalitat de les aplicacions.
C	Un grau extremadament alt de concurrència en cada agent implantat. No necessàriament. Els sistemes distribuïts, per definició, són concurrents però la concurrència pot ser obtinguda executant múltiples fils o processos. És a dir, no s'obliga al fet que cada agent siga un procés multi-fil. Podem implantar la concurrència mitjançant diversos processos amb un sol fil d'execució cadascun. Per exemple, en Node.js succeirà això.
D	Estar programats en Node.js. No. Es pot utilitzar qualsevol llenguatge de programació que resulte útil per a implantar la funcionalitat de cada component. Java, C i C++ són exemples típics. La distribució no depèn per a res del llenguatge de programació utilitzat.
E	Totes les anteriors.
F	Cap de les anteriors.

12. Alguns dels problemes fonamentals (i les seues solucions) en la computació distribuïda són...

A	Coordinació de components (via pas de missatges, dissenyant algorismes que requerisquen un intercanvi mínim de missatges).
----------	--

B	Gestió de fallades (utilitzant replicació, detectors de fallades i mecanismes de recuperació).
C	Persistència d'estat (via <i>commit</i> distribuït, replicació i emmagatzematge persistent).
D	Consistència d'estat (utilitzant replicació i protocols de consistència).
E	Totes les anteriors. El tema 2 presenta la coordinació, la gestió de fallades, la persistència d'estat i la consistència d'estat com quatre problemes fonamentals en la computació distribuïda. Algunes de les seues possibles solucions han sigut incloses en cadascun dels apartats anteriors, entre parèntesis.
F	Cap de les anteriors.

13. El model de sistema distribuït presentat en el Tema 2...


A	...considera detalls de baix nivell sobre el comportament del sistema. Així garanteix un resultat més precís en l'etapa de disseny del programari. Fals. Els detalls de baix nivell no es consideren en els models. Un model elimina aquests detalls i presenta una imatge d'un determinat sistema o mecanisme emprant un nivell d'abstracció alt.
B	...assumeix que tots els agents són multi-fil. Fals. El model assumeix que els agents són processos seqüencials, representats mitjançant autòmats. L'execució d'una acció es modela com una transició entre estats. En cada autòmat solament pot haver-hi una única seqüència de transicions; és a dir, representa un únic fil d'execució.
C	...sempre assumeix processos sincrònics i comunicació asincrònica. Fals. Per exemple, la comunicació pot ser asincrònica. Al Tema 2 es van definir diferents nivells de sincronia, considerant múltiples aspectes: comunicació, rellotges, canals, processos... De fet, se suggeria que la comunicació havia de ser asincrònica per a millorar l'escalabilitat.
D	...representa l'execució de processos com una seqüència d'accions o esdeveniments que poden interrompre's. Fals. Les accions han de ser atòmiques, és a dir, no poden ser interrompudes una vegada iniciades. Els termes esdeveniment i acció no són sinònims. Un esdeveniment es comporta com una guarda o preconditioni per a l'execució d'una acció. Això implica que l'acció no serà iniciada fins que ocórrega l'esdeveniment corresponent. No té sentit parlar de "esdeveniments que poden interrompre's".
E	Totes les anteriors.
F	Cap de les anteriors.

14. Quan comparem servidors asincrònics amb servidors multi-fil...

A	Els servidors asincrònics implanten de manera trivial les accions atòmiques definides en el model de sistema distribuït proposat en el Tema 2. Vertader. Els servidors asincrònics solen tenir un únic fil d'execució i usen esdeveniments com guardes o preconditionis per a l'inici de l'execució d'una acció. Les accions no poden interrompre's una vegada iniciades. Per tant, s'executen atòmicament.
---	--

B	<p>“Dirigit per esdeveniments” és un sinònim per a “multi-fil”.</p> <p>Fals. “Dirigit per esdeveniments” és un sinònim per a “servidor asincrònic”.</p>
C	<p>Els servidors asincrònics es bloquegen quan hi ha concurrència mentre els servidors multi-fil realitzen accessos concurrents a recursos sense bloquejar-se.</p> <p>Fals. Cada servidor asincrònic sol tenir un sol fil. Per tant, no poden bloquejar-se a causa d'interferències amb altres fils. Per contra, en un servidor multi-fil sí que necessitem bloquejar als fils quan intenten accedir a algun recurs compartit amb la resta de fils. Això ocorre amb les variables compartides, per exemple, per a implantar seccions crítiques amb la protecció adequada per a evitar inconsistències en les seues modificacions.</p>
D	<p>JavaScript és un exemple de llenguatge de programació específicament dissenyat per a implantar servidors multi-fil.</p> <p>Fals. És un exemple de llenguatge de programació que pot implantar servidors asíncrons.</p>
E	Totes les anteriors.
F	Cap de les anteriors.

15. Aquestes són algunes propietats que hem d'exigir als sistemes distribuïts...

A	Tolerància a fallades.
B	Alta disponibilitat.
C	Seguretat.
D	Escalabilitat.
	<p>Totes les anteriors.</p> <p>Cert. Aquestes quatre propietats han sigut llistades en la introducció del Tema 2 com a propietats a exigir en tot sistema distribuït.</p>
F	Cap de les anteriors.

16. La consistència d'estat significa que...

A	<p>L'estat gestionat per un component només pot tenir una única instància en el sistema; p. ex., està emmagatzemat en una base de dades centralitzada.</p> <p>Fals. La consistència d'estat és un sinònim per a “consistència entre rèpliques”. Aquest terme només té sentit quan hi haja múltiples còpies de cada element o dada. Aquest apartat afirma just el contrari. Assumeix que hi ha una sola instància en tot el sistema. En aquest cas, no hi haurà rèpliques. Per tant, no podríem parlar sobre la consistència d'estat o sobre el model de consistència utilitzat en aquest sistema.</p>
---	---

B	<p>Totes les variables globals haurien de ser accedides en exclusió mútua per a evitar condicions de carrera.</p> <p>Fals. Vegeu la justificació de l'apartat anterior.</p> <p>Les “condicions de carrera” també podran generar inconsistències, però aquestes inconsistències no estan relacionades amb la divergència entre les diferents rèpliques d'una dada en un sistema distribuït.</p> <p>En un model de programació amb servidors asincrònics (amb un sol fil d'execució cadascun) no serà necessari preocupar-se per les “condicions de carrera”. L'exclusió mútua en l'accés a les variables d'un determinat procés estarà garantida sense necessitat de prendre cap precaució addicional.</p>
C	<p>Quan un component està replicat hi ha un conjunt d'invariants que limiten el grau de divergència entre les rèpliques de cadascuna de les seues dades.</p> <p>Cert. Aquest és el significat utilitzat en el Tema 2 per a aquest concepte.</p>
D	<p>Quan un component està replicat, o bé totes les seues rèpliques estan actives i funcionen correctament o totes fallen i no poden continuar.</p> <p>Fals. No té excessiu sentit obligar al fet que totes les rèpliques d'un servei paren alhora. La replicació s'utilitza per a aconseguir la continuïtat dels serveis. Quan una rèplica falle, les restants han d'aconseguir que el servei segueixca estant disponible (és a dir, que el servei no falle).</p>
E	Totes les anteriors.
F	Cap de les anteriors.

17. La persistència d'estat significa que...

A	<p>Una aplicació distribuïda no pot tenir dades volàtils. Totes les dades haurien de residir en fitxers (en el disc dur) o en bases de dades.</p> <p>Fals. Quan es garantisca la persistència d'estat, assegurarem la durabilitat de les dades. No obstant això, això no prohibeix que utilitzem altres variables temporals. Per exemple, per a mantenir algunes dades que puguin calcular-se a partir de les que es guarden de manera persistent. Això pot ser interessant si les dades “derivades” han de ser accedides sovint. En lloc de calcular-les en cada petició, les mantindrem en algunes variables, en memòria principal.</p>
---	---

B	<p>L'accés a qualsevol dada hauria de realitzar-se sempre sota la protecció d'una transacció distribuïda.</p> <p>Fals. El terme “transacció distribuïda” es refereix a un conjunt de subtransaccions executades cadascuna en un node diferent. Totes les subtransaccions col·laboren en una seqüència global d'operacions sobre la qual s'asseguren les propietats d'atomicitat i aïllament. No és obligatori l'ús de transaccions distribuïdes per a accedir a dades emmagatzemades en memòria secundària. En molts casos bastarà amb executar una transacció local o, simplement, un accés aïllat a la dada d'interès.</p>
C	<p>Quan s'aplique un canvi sobre una dada persistent, la seua durabilitat està garantida.</p> <p>Cert. Est és el significat utilitzat en el Tema 2 per a referir-nos a la “persistència d'estat”.</p>
D	<p>Cada dispositiu d'emmagatzematge secundari que siga utilitzat per una aplicació distribuïda estarà replicat.</p> <p>Fals. La replicació no necessita aplicar-se sobre els dispositius d'emmagatzematge secundari per a garantir la durabilitat de la informació. Si existira aquesta obligació, tots els discos utilitzats en un sistema distribuït haurien de ser discs RAID (<i>Redundant Array of Inexpensive Disks</i>). No és la solució comuna.</p> <p>A part, la persistència d'estat no depèn de l'emmagatzematge secundari. Alguns SGBD d'altres prestacions mantenen les dades en memòria principal, utilitzant múltiples ordinadors amb molta memòria RAM. En aquest cas sí que resulta necessària la replicació per a assegurar la durabilitat. VoltDB és un exemple d'aquest tipus.</p>
E	Totes les anteriors.
F	Cap de les anteriors.

18. En el model de sistema distribuït del Tema 2...

A	<p>Els esdeveniments interns es refereixen a accions aplicades per la lògica de l'agent. Per exemple, per a rebre un missatge.</p> <p>Fals. Els esdeveniments interns s'han definit així, però la recepció d'un missatge no és un esdeveniment intern. És un esdeveniment extern. Està relacionat amb una acció que ha sigut iniciada per un altre agent.</p>
---	---

B	Tant els esdeveniments interns com els externs generen transicions d'estat. Cert. Els esdeveniments, de qualsevol tipus, generen transicions d'estat ja que inicien l'execució d'una acció.
C	L'execució d'un agent es modela com una única seqüència d'esdeveniments. Tant la concurrència com els agents multi-fil no poden representar-se. Fals. La primera frase és certa, però la segona oració és falsa. Encara que els agents multi-fil no estan suportats per aquest model, això no implica que es renuncie a la concurrència. Un sistema distribuït està compost per múltiples agents. Malgrat que cadascun siga un agent seqüencial, l'escenari global és concurrent. Els agents compartiran recursos. Col·laboraran per a aconseguir un objectiu comú. Hi haurà interacció entre ells. Amb això, formaran un sistema concurrent. Per tant, la concurrència està representada i és admesa per aquest model de sistema.
D	Ja que tots els sistemes distribuïts han de ser transparents davant fallades, aquest model assumeix que les fallades mai ocorren. Fals. L'especificació donada en aquest model assumeix que les fallades podran ocórrer. El tipus de fallada assumida és la parada dels processos.
E	Totes les anteriors.
F	Cap de les anteriors.

19. La comunicació en el model de sistema senzill del Tema 2...

A	...assumeix que els esdeveniments interns defineixen una relació "precedeix localment" d'ordre total en cada agent. Cert. Aquesta és la condició FIFO inclosa en la relació causal.
---	--

B	<p>...assumeix que els esdeveniments externs defineixen una relació “causa directament” on un esdeveniment d'eixida és la causa d'un esdeveniment d'entrada.</p> <p>Cert. L'esdeveniment d'eixida més comú és l'enviament d'un missatge. L'esdeveniment d'entrada típic és la recepció d'un missatge. La transmissió de missatges està modelada per la condició “causa directament” de la relació causal.</p>
C	<p>El tancament transitiu de les relacions “causa directament” i “precedeix localment” defineix la relació de comunicació “causal”.</p> <p>Cert. Lamport va definir la relació “happens before” (o relació causal) com el tancament transitiu de dues condicions elementals (seqüència interna d'esdeveniments o “precedeix localment” i esdeveniments relacionats per la transmissió d'un missatge o “causa directament”). Aquestes dues condicions reben en el Tema 2 els noms esmentats en aquest apartat.</p>
D	<p>La relació de comunicació “causal” permet identificar als esdeveniments no relacionats com a “concurrents”.</p> <p>Cert. La relació causal estableix dependències entre els esdeveniments. Quan un parell d'esdeveniments no està relacionat (és a dir, no es compleix “$a \rightarrow b$” ni “$b \rightarrow a$”), llavors direm que els dos esdeveniments són concurrents (és a dir, “$a \parallel b$”).</p>
E	Totes les anteriors.
F	Cap de les anteriors.

20. Per a especificar programes en el model de sistema senzill del Tema 2...

A	<p>El model assumeix guardes atòmiques, protegides per accions.</p> <p>Fals. El model assumeix accions atòmiques protegides per guardes.</p>
---	--

B	<p>Les accions atòmiques són una font potencial d'errors. Per això, estan implantades com a blocs de codi que pot interrompre's en tots els llenguatges de programació.</p> <p>Fals. Les accions atòmiques eviten les "condicions de carrera". Per tant, són una eina que evita molts errors a l'hora d'executar un programa. L'atomicitat ha de ser acuradament mantinguda quan el nostre model de sistema passe a ser implantat sobre un sistema real. El llenguatge de programació utilitzat ha de facilitar eines per a mantenir l'atomicitat. Això implica que cada fragment de codi que implante una acció no pot ser interromput lògicament.</p>
C	<p>Les guardes són una font potencial de condicions de carrera. Per això no s'utilitzen en els llenguatges de programació multi-fil.</p> <p>Fals. Les guardes i les accions atòmiques eviten les "condicions de carrera". Això ja s'ha comentat en l'apartat anterior. Quan el llenguatge de programació suporta múltiples fils d'execució, les accions atòmiques han de traduir-se com a operacions d'un monitor i les guardes serien les condicions que permeten accedir a (o continuar amb l'execució de) aquestes operacions. Aquesta aproximació ja va ser estudiada en CSD.</p>
D	<p>El model assumeix accions atòmiques protegides per condicions (també anomenades guardes).</p> <p>Cert. Això ja s'ha indicat en el primer apartat i s'ha explicat en l'apartat C.</p>
E	Totes les anteriors.
F	Cap de les anteriors.

21. El middleware és una capa de programari que...

A	<p>...està col·locat entre el maquinari i el sistema operatiu.</p> <p>Fals. Està situat sobre el sistema operatiu, facilitant cert grau de transparència de distribució a les aplicacions desenvolupades sobre ell.</p>
B	<p>...garanteix transparència de fallades per als components de les aplicacions distribuïdes.</p> <p>Fals. S'han presentat diferents tipus de middleware en el Tema 3. La majoria no gestionen les fallades ni suporten directament la replicació de components. Per tant, són incapaços de garantir la transparència de fallades. Per exemple, la RPC (crida a procediment remot) és un exemple de middleware. Facilita transparència d'ubicació, però no replica automàticament els serveis invocats. Per tant, quan un servidor falle utilitzant RPC, el seu client obtindrà una excepció com a resultat de la fallada.</p>
C	<p>...utilitza contenidors per a desplegar serveis distribuïts.</p> <p>Fals. Un middleware no sempre haurà de preocupar-se del desplegament de components. RPC i RMI són dos exemples de middleware que no ho fan.</p>
D	<p>...està implantat en JavaScript.</p> <p>Fals. Els middleware faciliten interoperabilitat. El llenguatge de programació utilitzat per a programar els seus elements no cal que siga JavaScript.</p>
E	Totes les anteriors.
F	Cap de les anteriors.

22. Algunes característiques del middleware són...

A	Proporciona una API estàndard.
B	Utilitza protocols d'interacció estàndard.
C	Proporciona serveis d'interès general.
D	Garanteix la interoperabilitat de components desplegats sobre diferents plataformes.
E	Totes les anteriors. Totes aquestes característiques s'han llistat i explicat en la guia d'estudi del Tema 3. Aquestes característiques van ser descrites i proposades en l'article escrit per Phil Bernstein en el qual es va analitzar què havia d'entendre's com middleware i es van classificar els diferents tipus de middleware existents.
F	Cap de les anteriors.

23. Els sistemes d'objectes distribuïts...

A	...necessiten un middleware per a gestionar la invocació a objectes remots. Cert. S'utilitza un <i>Object Request Broker (ORB)</i> per a aquesta fi. És un tipus particular de middleware.
B	...són inherentment menys escalables que els sistemes distribuïts basats en sistemes de missatgeria. Cert. Usen una interacció client-servidor. Això implica un model de comunicació síncron (similar al REQ/REP de ZeroMQ) que bloqueja al client fins que una resposta és enviada pel servidor. Tota font de bloquejos hauria de ser evitada per a desenvolupar sistemes distribuïts escalables.
C	...tenen un major acoblament que els sistemes distribuïts no orientats a objectes. Cert. Els sistemes d'objectes distribuïts utilitzen referències a objecte per a accedir als objectes remots. En molts casos, això permet que es transferisca una bona quantitat d'informació entre diferents components del sistema en cada invocació. En aquesta situació es necessitarien missatges grans i es reduiria el rendiment del sistema. Una alta taxa de missatges enviats i rebuts entre components que interactuen entre si seria un indicatiu d'un acoblament alt. Això ha d'evitar-se en els sistemes escalables.
D	...ofereixen, normalment, transparència d'ubicació. Sí. L'ús de <i>proxies</i> en els clients proporciona transparència d'ubicació. Els processos clients no saben en quin ordinador resideix l'objecte que s'està invocant.
E	Totes les anteriors.
F	Cap de les anteriors.

24. Els sistemes de missatgeria...

A	...són persistents quan l'emissor roman bloquejat esperant alguna resposta del
---	--

	receptor. Fals. Són sincrònics en aquest cas.
B	...són no persistents quan la comunicació està dirigida per un agent gestor (<i>broker</i>). Fals. Els gestors (<i>brokers</i>) solament es necessiten en la comunicació persistent.
C	...poden ser persistents i estar basats en gestor (<i>broker-based</i>). Cert. Aquesta és una combinació vàlida. L'estàndard AMQP (<i>Advanced Message Queueing Protocol</i>) utilitza aquesta aproximació.
D	...poden ser sincrònics i no persistents. ZeroMQ is un exemple d'aquest tipus. Fals. La combinació és vàlida i arriba a ser utilitzada en alguns casos. Per exemple, RPC és un exemple ben conegut. No obstant això, ZeroMQ no utilitza aquesta combinació: és feblement persistent (en lloc de "no persistent"), sense gestor (<i>brokerless</i>) i asincrònic.
E	Totes les anteriors.
F	Cap de les anteriors.

25. Els estàndards...

A	...faciliten la interoperabilitat. Cert. Aquest és un dels seus principals objectius.
B	...no poden utilitzar-se en sistemes distribuïts. Fals. Els middleware implanten estàndards i s'utilitzen en els sistemes distribuïts.
C	...garanteixen la transparència de fallades. Fals. No tots els estàndards són específics dels sistemes distribuïts. Sense utilitzar components replicats no es podran gestionar alguns escenaris de fallada. Per tant, fins i tot els estàndards utilitzats en sistemes distribuïts podran en alguns casos no gestionar adequadament les fallades.
D	...milloren el rendiment. Fals. Molts estàndards faciliten solucions eficients a certs problemes, però pot haver-hi altres solucions no estàndard amb millor rendiment. Per tant, no tots els estàndards faciliten la solució òptima des del punt de vista del rendiment.
E	Totes les anteriors.
F	Cap de les anteriors.

26. Des del punt de vista d'un programador, quan se segueix un estàndard...

A	...els programes són fàcils d'escriure ja que els elements utilitzats ofereixen una menor complexitat.
---	--

B	...el resultat final és més fiable ja que l'estàndard defineix clarament com fer les coses.
C	...el codi té un manteniment senzill perquè, encara que els estàndards s'actualitzen, els seus canvis solen mantenir la interoperabilitat.
D	...els programes són fàcils d'escriure ja que els estàndards estan basats en conceptes clars i ben definits.
E	Totes les anteriors. Cert. Tots aquests aspectes van ser considerats quan es van descriure els estàndards en el Tema 3. L'estandardització i la interoperabilitat són dos dels principals objectius dels <i>middleware</i> .
F	Cap de les anteriors.

27. Dues aproximacions d'invocació a mètodes remots en l'àrea de serveis web són...

A	SOAP i REST. Cert. Aquestes han sigut les dues aproximacions presentades amb aquest objectiu en el Tema 3. La guia d'estudi explicava detalladament la segona. SOAP es descriurà en altres assignatures.
B	ZeroMQ i nanomsg. Fals. Són exemples de middleware de comunicacions basats en missatges que implanten comunicació asincrònica.
C	RPC i RMI. Fals. Són mecanismes d'invocació remota utilitzats en sistemes orientats a objectes o basats en processos. Cap dels dos ha sigut específicament dissenyat per a ser utilitzat en serveis web.
D	Client-servidor i <i>peer-to-peer</i> . Fals. Són les dues aproximacions genèriques d'interacció entre agents en la computació distribuïda. La interacció peer-to-peer no està basada en invocació de mètodes o operacions.
E	Totes les anteriors.
F	Cap de les anteriors.

28. L'estil arquitectònic REST...

A	...usa HTTP com a "transport".
B	...usa només quatre mètodes "bàsics": GET, PUT, POST i DELETE.
C	...usa el seu mètode GET per a accions de només lectura.
D	Pren l'estil arquitectònic client-servidor com a base i promou l'ús de servidors "stateless" (per a gestionar fàcilment les fallades).
E	Totes les anteriors. Aquestes són les principals característiques de l'estil arquitectònic REST. Van ser descrites detalladament en la guia d'estudi del Tema 3.
F	Cap de les anteriors.

29. Alguns exemples d'"un altre middleware" són...

A	gedit. Fals. És un editor de text convencional dels entorns X-Window. Com a tal, és solament un exemple d'aplicació de nivell d'usuari. Tampoc assumeix que haja
---	---

	de ser executada en un entorn distribuït.
B	OAuth. Cert. És un exemple de <i>middleware</i> relacionat amb seguretat.
C	Linux. Fals. És un exemple de sistema operatiu.
D	MS-DOS. Fals. És un exemple de sistema operatiu.
E	Totes les anteriors.
F	Cap de les anteriors.

30. El middleware de nomenat...

A	...garanteix transparència de fallades. Fals. El servei de noms no participa de manera directa en cap subsistema de gestió de fallades. Encara que assumim que col·labore en un subsistema d'aquest tipus, no serà el seu component principal. Per tant, és incapaç d'assegurar transparència de fallades per si mateix.
B	...proporciona transparència d'ubicació. Cert. És un component clau per a assegurar transparència d'ubicació. Amb la seua ajuda els clients poden referir-se a agents externs (servidors) utilitzant un nom. Aquests noms són traduïts a adreces pel middleware de nomenat. Així, l'adreça real no és coneguda pels clients. Això proporciona transparència d'ubicació. Tant DNS com el servei de noms dels sistemes CORBA són exemples d'aquest tipus de serveis.
C	...implementa servidors sense estat (<i>stateless</i>). Fals. Que els serveis siguin <i>stateless</i> o <i>stateful</i> no depèn de la utilització d'un middleware de nomenat.
D	...millora l'escalabilitat del sistema. Fals. La resolució de noms introdueix un pas addicional (que normalment requerirà comunicació amb una entitat remota) a l'hora d'accedir a un objecte, servei o procés referenciat mitjançant un nom. Per això, encara que els serveis de noms són convenients en facilitar transparència d'ubicació, els seus resultats han de mantenir-se en memòria per a no perjudicar excessivament al rendiment i a l'escalabilitat del sistema.
E	Totes les anteriors.
F	Cap de les anteriors.

31. Un Acord de Nivell de Servei (o SLA) és...

A	...un acord entre proveïdors de servei i clients de servei.
B	...una especificació de característiques de servei (p. ex., funcionalitat, temps de resposta, rendiment, disponibilitat...) i els seus nivells a garantir.
C	...un dels aspectes a considerar per a decidir el nombre d'instàncies de cada component durant el desplegament d'un servei.
D	...alguna cosa a considerar en sistemes PaaS per a reomplir el pla de desplegament i el pla d'escalat d'un servei determinat.
E	Totes les anteriors. Tots els apartats anteriors són certs. Aquesta informació es va facilitar en les diferents seccions del Tema 4. El SLA és un acord establert entre proveïdors i clients d'un servei. En un escenari ideal, l'acord ha d'especificar la qualitat de servei que oferirà el proveïdor així com la màxima càrrega introduïda per cada client. Aquesta informació s'utilitza durant l'etapa de desplegament per a gestionar els aspectes descrits en els apartats C i D.
F	Cap de les anteriors.

32. En el Tema 4, un servei és...

A	...una aplicació distribuïda que ha sigut desplegada i roman activa. Cert. Aquesta ha sigut la definició informal proporcionada en la introducció del Tema 4.
B	...un conjunt de <i>scripts</i> independents amb un pla de desplegament. Fals. En el cas general un servei està format per múltiples components que col·laboren per a proporcionar un objectiu comú. No poden ser independents, sinó estar estretament relacionats. Res ens obliga a implantar-los com <i>scripts</i> .
C	...una futura aplicació distribuïda que encara està en les seues etapes d'anàlisi o disseny. Fals. Un servei és una aplicació distribuïda que ja està en execució; d'una altra manera no podria utilitzar-se. Per això, ja ha superat les etapes d'anàlisi i disseny en el seu cicle de vida.
D	...un programa Node.js que és executat per un únic usuari. Fals. En tal cas el programa no necessitaria un entorn distribuït. En el context del Tema 4, el terme "servei" es refereix a serveis de programari distribuïts .
E	Totes les anteriors.
F	Cap de les anteriors.

33. Aquestes són algunes tasques a considerar quan una aplicació distribuïda està sent desplegada...

A	Decidir quantes instàncies de cada component haurien de ser executades i on.
B	Decidir quins serveis dependents haurien de ser utilitzats per aquesta aplicació distribuïda.
C	Decidir l'ordre en què cadascun dels seus components hauria de ser iniciat.
D	Contactar amb el sistema operatiu o contenidor en cada amfitrió perquè iniciï els seus components.
E	Totes les anteriors. Cert. Totes aquestes tasques són exemples de passos que han de ser tinguts en compte.
F	Cap de les anteriors.

34. L'administració del cicle de vida d'un servei està estretament relacionada amb el desplegament. Algunes de les seues tasques són...

A	Actualització de components.
----------	------------------------------

B	Canvis de configuració.
C	Detecció i recuperació de fallades en els components.
D	Decisions d'escalat, depenent de la càrrega actualment suportada.
E	Totes les anteriors. Cert. De nou, totes aquestes tasques formen part de la gestió del cicle de vida d'un servei. Van ser descrites dins del Tema 4.
F	Cap de les anteriors.

35. Alguns problemes que sorgeixen quan una aplicació tradicional és desplegada en un ordinador de sobretaula són...

A	Resolució de dependències del programari; és a dir, trobar les biblioteques apropiades de les quals depenga l'aplicació.
B	Donar valors apropiats a les variables d'entorn utilitzades per l'aplicació, si les hi hagués.
C	Configurar adequadament l'aplicació (p. ex., via registre en Windows, arxius de configuració en Linux, fitxers en <code>/Library</code> en Mac OS, etc.)
D	Esbrinar si els requisits d'instal·lació de l'aplicació estan suportats per l'estat actual de l'ordinador amfitrió i el seu sistema operatiu.
E	Totes les anteriors. Cert. Totes les afirmacions anteriors proporcionen exemples dels aspectes a considerar en desplegar una aplicació en un ordinador convencional. L'apartat C sol ser gestionat pel programa instal·lador i resulta transparent per a l'usuari.
F	Cap de les anteriors.

36. Alguns dels elements en un descriptor de desplegament són...

A	Middleware de nomenat. Fals. No té sentit incloure un middleware (el que siga) en un descriptor de desplegament.
---	---

B	Ordre a utilitzar per l'usuari (p. ex., docker). Fals. El descriptor de desplegament dirigeix les tasques del desplegament. L'ordre a utilitzar per l'usuari es va explicar en el context del Seminari 6, en descriure els contenidors. No forma part del descriptor de desplegament.
C	Pla de desplegament. Cert. Les principals parts d'un descriptor de desplegament són un pla de desplegament, les plantilles de configuració de components ja emplenades i les descripcions de dependències.
D	Dockerfile. Fals. El "Dockerfile" és un fitxer de configuració específic, necessari quan s'utilitzen contenidors <i>docker</i> . S'utilitza per a proporcionar la seqüència d'instruccions necessària per a construir una imatge <i>docker</i> . No forma part dels descriptors de desplegament "normals".
E	Totes les anteriors.
F	Cap de les anteriors.

37. Un component necessita els elements següents per a ser desplegat...

A	El seu programa (o BLOB).
B	Una plantilla de configuració reomplerta.
C	Una descripció de totes les seues dependències.
D	Una especificació del seu "endpoint".
E	Totes les anteriors. Cert. Tots aquests elements es necessiten per a desplegar un component. Els apartats B, C i D són parts habituals en els descriptors de desplegament. Sense l'apartat A no es podria desplegar res. És precisament l'element a desplegar.
F	Cap de les anteriors.

38. La injecció de dependències...


A	...desacobla el codi del component de qualsevol implementació concreta de dependències, i està suportada en els entorns de contenidors. Cert. Aquesta és la definició d'injecció de dependències utilitzada en el Tema 4.
---	--

B	...requereix l'ús de variables d'entorn per a resoldre dependències. Fals. Quan s'utilitza la injecció de dependències, aquestes es resoldran "incrustant" els mòduls apropiats en el component durant la fase de desplegament. Amb aquest tipus de resolució de dependències, les variables d'entorn pot ser que no es necessiten.
C	...requereix l'ús d'arxius de configuració per a resoldre dependències. Fals. Ocorre el mateix que en el cas anterior.
D	...soluciona totes les dependències estàticament, és a dir, durant la implementació. Fals. La injecció de dependències és un mecanisme que permet un enllaç tardà entre components, durant la fase de desplegament generalment. Decidir aquest enllaç mentre s'escriu el programa, amb un mecanisme estàtic, és poc aconsellable. Si hi haguera algun canvi en els components a utilitzar, això ens obligaria a no poder resoldre les dependències, amb el que no es podrien desplegar els components en aquest tipus de resolució.
E	Totes les anteriors.
F	Cap de les anteriors.

39. En el model de servei IaaS...

A	...el desplegament està completament automatitzat pel proveïdor. Fals. Això solament ocorreria en el model de servei PaaS i encara no ha aconseguit un grau suficient de maduresa.
B	...vàries decisions de desplegament inicials no estan automatitzades: quantitat d'instàncies per component, tipus de MV requerit per cada component... Vertader. Aquestes decisions han de ser preses pel client (que és qui intenta desplegar una aplicació) en aquest model de servei. Observe's que el client no és l'usuari final de les aplicacions, sinó el seu desenvolupador.
C	...les decisions de desplegament relacionades amb el cicle de vida estan automatitzades; p. ex., quins nivells de càrrega disparen accions d'escalat, com s'actualitza el programari d'un component... Fals. Ocorre el mateix que en l'apartat A.
D	...cap suport per al desplegament és gestionat pel proveïdor. Fals. Un suport mínim sí que s'arriba a facilitar. Per exemple, per a desplegar una imatge sobre una màquina virtual.
E	Totes les anteriors.
F	Cap de les anteriors.

40. En Windows Azure, alguns aspectes del seu suport de desplegament són...

A	Hi ha un pla bàsic d'actualització de serveis, a pesar que no suporta serveis amb estat.
B	Els components es diuen "rols".
C	Hi ha una administració bàsica de dominis de fallada que millora la disponibilitat de servei.
D	No hi ha cap pla de seqüenciació de desplegament.
	Totes les anteriors. Cert. Tots aquests aspectes van ser inclosos en la descripció de Windows Azure proporcionada en l'última secció del Tema 4.
F	Cap de les anteriors.