Este examen consta de 40 cuestiones de opción múltiple. En todas las cuestiones hay una única opción correcta. Las respuestas deben proporcionarse en la hoja de respuestas que se ha proporcionado separada de este enunciado.

Todas las cuestiones tienen el mismo valor. Las que estén correctamente contestadas aportarán 0.25 puntos a la nota de este parcial. Las incorrectas descontarán 0.05 puntos. En caso de duda, se recomienda dejarlas en blanco.

Dispone de dos horas para completar el examen.

1. El acoplamiento en una aplicación distribuida:

A	Debe ser alto, pues así se reduce la necesidad de comunicación y las aplicaciones ofrecen mejor rendimiento. FALSO. El acoplamiento mide las necesidades de comunicación entre módulos, examinando el número de argumentos necesarios en cada interacción así como los tipos y ámbito de esos argumentos. Se recomienda que el acoplamiento sea lo más bajo posible.
B	No existe. En una aplicación distribuida sólo hay cohesión, jamás habrá acoplamiento. FALSO. Al poder distinguir múltiples componentes en cualquier aplicación distribuida, el acoplamiento estará referido a las necesidades de comunicación entre esos módulos o componentes. Sí que tiene sentido hablar de acoplamiento en estos entornos.
C	Debe ser lo más bajo posible. Así sólo se transmitirá la información estrictamente necesaria en cada interacción. CIERTO. Ya se ha explicado el porqué en los dos apartados anteriores.
D	Está siempre ligado a la cohesión. Cuanto más bajo sea el acoplamiento más baja será la cohesión. FALSO. Generalmente se necesita una cohesión alta para obtener un acoplamiento débil.
E	Todas las anteriores. La única opción correcta es la C.
F	Ninguna de las anteriores. La única opción correcta es la C.

2. La cohesión en cualquier aplicación distribuida:

A	Está ligada al acoplamiento. Cuanto más baja sea la cohesión, más bajo será el acoplamiento. FALSO. Generalmente se necesita una cohesión alta para obtener un acoplamiento débil.
B	Conviene que sea lo más alta posible. Así será fácil reutilizar los módulos desarrollados. CIERTO. Cuando la cohesión es alta, cada módulo de una aplicación realiza una única función y está destinado en exclusiva a ello. Si posteriormente se necesitara esa misma funcionalidad en otras aplicaciones y el diseño de ese módulo fue cuidadoso, resultará fácil reutilizar el módulo.
C	Conviene que sea lo más baja posible. Así se reduce la necesidad de comunicación entre nodos. FALSO. Con una cohesión baja se habrán desarrollado múltiples módulos en una aplicación que desarrollarán múltiples funciones cada uno. Eso refleja un mal diseño. Si el diseño fue de baja calidad nada garantizará que se haya buscado reducir las necesidades de comunicación entre módulos. Además, aquello que podrá reducir las necesidades de comunicación es el acoplamiento bajo. La cohesión guarda relación con la funcionalidad de cada módulo, no con sus necesidades de comunicación.
D	Conviene que sea lo más alta posible. Así mejora la robustez de la aplicación. FALSO. La robustez (fiabilidad, disponibilidad, mantenibilidad y seguridad) es una colección de atributos no funcionales de las aplicaciones y no guarda relación directa con el grado de cohesión de una aplicación, que sí depende de la funcionalidad de cada módulo.
E	Todas las anteriores.
F	Ninguna de las anteriores.

3. La persistencia en una aplicación distribuida…:

A	No es recomendable, pues reduce el rendimiento e impide que las aplicaciones sean escalables. FALSO. En cualquier aplicación distribuida se puede necesitar que algunos datos se mantengan de modo persistente. Si se modifican con baja frecuencia o se realiza un reparto adecuado de ellos a la hora de gestionarlos no tiene por qué limitarse fuertemente el rendimiento de las aplicaciones. Por ejemplo, MongoDB es un ejemplo de almacén escalable de datos persistentes.
B	Permite superar algunas situaciones de fallo sin que se pierda por completo el estado. CIERTO. Los datos almacenados de manera persistente superan varios tipos de fallos en los ordenadores que los gestionan (de hecho, todos aquellos fallos que no impliquen una destrucción del medio físico donde residan). Esos datos pueden utilizarse localmente tan pronto como el nodo se recupere. En muchos casos esto permitirá reanudar el servicio sin requerir la colaboración de otros nodos o minimizando la cantidad de información a transferir para completar la reconfiguración, agilizando su recuperación.
C	Conduce a una alta cohesión. FALSO. La persistencia no guarda relación con la cohesión.
D	Reduce el acoplamiento. FALSO. La persistencia no guarda relación con el acoplamiento.
E	Todas las anteriores.
F	Ninguna de las anteriores.

4. Las bases de datos relacionales:

A	Soportan transacciones con garantías ACID. CIERTO. Las garantías de atomicidad, consistencia, aislamiento y durabilidad son proporcionadas por las transacciones utilizadas en los sistemas gestores de bases de datos relacionales.
B	Proporcionan persistencia. CIERTO. La persistencia resulta necesaria para garantizar la durabilidad de las escrituras efectuadas en cada transacción.
C	Limitan la escalabilidad, pues conducen a bloqueos frecuentes si hay un elevado grado de concurrencia. CIERTO. El control de concurrencia automatizado que se utiliza en los sistemas gestores de bases de datos relacionales puede provocar bloqueos apreciables cuando el grado de concurrencia sea alto. Es el precio a pagar para obtener un aislamiento fuerte ("serializable") entre las transacciones.
D	Minimizan la duplicación de datos en sus tablas. Para ello utilizan técnicas de normalización. CIERTO. El objetivo de la normalización es reducir la cantidad de veces que se mantiene un campo de una entidad determinada en las relaciones (tablas) de una base de datos.
E	Todas las anteriores.
F	Ninguna de las anteriores.

5. Las garantías ACID de las transacciones son ...

A	Aislamiento, concurrencia, internacionalización y durabilidad.
B	Aislamiento, concurrencia, indexación y disponibilidad.
C	Atomicidad, concurrencia, aislamiento y disponibilidad.
D	Fiabilidad, seguridad, mantenibilidad y disponibilidad.
E	Todas las anteriores.
F	Ninguna de las anteriores. Las garantías ACID de las transacciones son: atomicidad, consistencia semántica, aislamiento y durabilidad. Ninguno de los apartados anteriores citaba las cuatro garantías correctas.

6. Para mejorar la escalabilidad de un almacén de datos persistente mediante una base NoSQL:

A	Se utiliza una base de datos relacional. FALSO. Las bases de datos relacionales utilizan SQL como su lenguaje de interrogación. Por tanto, no formarían parte de la familia de "bases NoSQL".
B	Se renuncia al uso de transacciones con garantías ACID. CIERTO. Como ya hemos visto en cuestiones anteriores la persistencia no siempre limitará la escalabilidad. Por tanto, ¿qué la limita en una base de datos tradicional? Principalmente sus mecanismos de control de concurrencia. Esos mecanismos resultan necesarios para garantizar el aislamiento entre transacciones concurrentes. Por tanto, si eliminamos las transacciones tradicionales el sistema resultante podrá ser más escalable que un sistema gestor relacional.
C	No se utiliza "*sharding*". FALSO. El "sharding" es uno de los mecanismos clave para mejorar la escalabilidad de un almacén de datos persistente.
D	Se utilizan lenguajes de interrogación complejos, permitiendo múltiples "joins". FALSO. El uso de "joins" en el lenguaje de interrogación complicaría excesivamente el reparto de los datos entre múltiples nodos (es decir, el "sharding"). Sin reparto de datos, la escalabilidad estará limitada. Además, la gestión de un "join" implicará que se acceda a múltiples tablas o colecciones y que se crucen los datos de varias de ellas. Eso ralentizaría excesivamente el servicio de ese tipo de sentencias, limitando también así la escalabilidad.
E	Todas las anteriores.
F	Ninguna de las anteriores.

7. MongoDB:

A	Es una base de datos relacional que utiliza "*sharding*" para mejorar su escalabilidad. FALSO. MongoDB utiliza "sharding" para mejorar su escalabilidad pero no es una base de datos relacional sino un almacén de documentos escalable.
B	Es un almacén de datos clave-valor. FALSO. MongoDB no pertenece a esta clase de almacenes de datos escalables.
C	Es un almacén de documentos. CIERTO. Esta es la clase de almacén a la que pertenece.
D	Es un almacén de registros extensibles. FALSO. MongoDB no pertenece a esta clase de almacenes de datos escalables.
E	Todas las anteriores.
F	Ninguna de las anteriores.

8. El teorema CAP establece…:

A	Que ningún sistema distribuido puede ser robusto.
B	Que un sistema no puede ser a la vez fiable, seguro y altamente disponible.
C	Que todos los sistemas distribuidos son escalables.
D	Que ningún sistema distribuido robusto admite situaciones de particionado de la red.
E	Todas las anteriores.
F	Ninguna de las anteriores. El teorema CAP establece que en cualquier sistema distribuido no se podrán garantizar simultáneamente tres propiedades: consistencia fuerte, disponibilidad de servicio y tolerancia a las situaciones de particionado de la red. Ninguno de los apartados anteriores decía eso. Quizá el apartado que haya podido sugerir más dudas sea el D. Un sistema distribuido robusto será simultáneamente fiable, disponible, mantenible (invertirá poco tiempo en su recuperación) y seguro. La robustez no exige nada en cuanto a la consistencia mínima que deberán respetar las réplicas de un determinado servicio. Por definición, un sistema robusto es altamente disponible. Podrá tolerar las situaciones de particionado de la red relajando temporalmente la consistencia entre las diferentes réplicas de cada componente. Con ello la afirmación del apartado D sería falsa.

9. Según el teorema CAP…

A	Un servicio altamente disponible no podrá mantener una consistencia fuerte cuando la red se particione y varias de sus réplicas queden desconectadas. CIERTO. Mantenemos la disponibilidad (A) y la tolerancia a situaciones de particionado (P) sacrificando la consistencia (C).
B	Cuando la red se particione los servicios del sistema seguirán estando disponibles y mantendrán una consistencia secuencial. FALSO. Como la consistencia secuencial es un modelo de consistencia fuerte se estarían manteniendo las tres propiedades (C, A y P). El teorema CAP implica que al menos una de ellas no habrá manera de garantizarla.
C	La red jamás podrá particionarse en un servicio escalable. FALSO. Para garantizar la disponibilidad asociada a un servicio escalable se suele preferir sacrificar la consistencia y admitir las situaciones de particionado.
D	La disponibilidad de un servicio se perderá cuando éste deba actualizarse. FALSO. Que se pierda la disponibilidad de un servicio cuando éste se actualice no tiene nada que ver con el teorema CAP.
E	Todas las anteriores.
F	Ninguna de las anteriores.

10. En un almacén de documentos…

A	El esquema de la base de datos es fijo y está formado por un conjunto determinado de tablas. FALSO. Eso ocurre en un sistema relacional, pero no en un almacén de documentos escalable.
B	El esquema de la base de datos es dinámico y está compuesto por una serie de colecciones capaces de albergar cada una cualquier clase de documento/objeto. CIERTO. Lo hemos podido comprobar al utilizar MongoDB.
C	Se utiliza SQL como lenguaje de interrogación. FALSO. Eso ocurre en los sistemas relacionales.
D	Las transacciones engloban múltiples sentencias y las sentencias admiten "joins" entre tablas o colecciones. FALSO. Eso ocurre en los sistemas relacionales.
E	Todas las anteriores.
F	Ninguna de las anteriores.

11. La seguridad…

A	Es un atributo de los sistemas distribuidos robustos. CIERTO. Junto a la fiabilidad, disponibilidad y mantenibilidad.
B	Es una propiedad derivada del teorema CAP. FALSO. El teorema CAP relaciona otras tres propiedades (consistencia, disponibilidad, tolerancia a las situaciones de particionado de la red) pero no dice nada sobre la seguridad.
C	Es una propiedad que cumple todo servicio distribuido altamente disponible. FALSO. La disponibilidad no implica seguridad. Si lo hiciera, no se habrían citado ambas explícitamente al definir la robustez.
D	Es una de las garantías de las transacciones ACID. FALSO. ACID acoge atomicidad, consistencia semántica, aislamiento y durabilidad, pero no cita para nada la seguridad.
E	Todas las anteriores.
F	Ninguna de las anteriores.

12. Objetivos principales de la seguridad:

A	Consistencia, disponibilidad y tolerancia a las particiones de la red. FALSO. Esas son las tres propiedades consideradas en el teorema CAP pero no guardan relación directa con los objetivos principales de la seguridad.
B	Disponibilidad, fiabilidad y mantenibilidad. FALSO. Estos son los tres otros atributos de la robustez ("dependability"), pero no pueden considerarse objetivos de la seguridad.
C	Recuperación, alta cohesión y bajo acoplamiento. FALSO. Son características aconsejables en cualquier sistema, pero no objetivos de la seguridad.
D	Confidencialidad, integridad, disponibilidad y contabilidad. CIERTO. Así se citaba en el tema 7.
E	Todas las anteriores.
F	Ninguna de las anteriores.

13. Los tres elementos que intervienen en una política de seguridad son:

A	Estrategias, mecanismos y garantías. FALSO. En cualquier sistema de seguridad habrá estrategias, mecanismos, políticas y garantías. Estos tres últimos elementos se consideran "herramientas de seguridad". Por tanto, los elementos citados en este apartado no intervienen en una política sino que son "elementos" que están a su mismo "nivel".
B	Agentes, objetos y acciones. CIERTO. El agente es el elemento que podrá realizar las acciones de una política. El objeto es el elemento sobre el que podrá actuar el agente. Por último, la acción es la operación que podrá llevar a cabo el agente sobre el objeto. Como resultado, una política es una especificación precisa que indica para cada agente qué acciones podrá realizar sobre cada objeto. Por tanto, esos son los tres elementos que intervienen en la política.
C	Vulnerabilidades, amenazas y ataques. FALSO. Esos son los tres tipos de riesgos que se han identificado al analizar la seguridad. Son tipos de riesgo, no elementos de una política.
D	Disponibilidad, consistencia y confidencialidad. FALSO. Tanto la disponibilidad de acceso como la confidencialidad son objetivos generales de la seguridad. La consistencia es una propiedad que debe buscarse en los sistemas en los que se replique información o servicios. Ninguno de los tres es un elemento que intervenga en una política de seguridad.
E	Todas las anteriores.
F	Ninguna de las anteriores.

14. El control de acceso:

A	Es un mecanismo de seguridad. CIERTO. Junto a los mecanismos físicos y los mecanismos de autenticación, es una de las tres clases de mecanismos de seguridad citados en el tema 7.
B	En caso de ser discrecional puede implantarse mediante listas de control de acceso o mediante capacidades. CIERTO. Para representar un control de acceso se suele utilizar una matriz de acceso en la que las filas representarán a los agentes y las columnas a los objetos. Cada componente de la matriz contendrá las acciones (o derechos) admitidas para esa combinación de agente y objeto. Como estas matrices suelen ser dispersas (es decir, tienen un alto número de componentes vacías) se implantan por columnas o por filas, utilizando solo las componentes no vacías. Al implantarse por columnas se genera una lista de control de acceso asociada a cada columna/objeto. Esto es lo que ocurre en los sistemas de ficheros. Las palabras de protección utilizadas en los sistemas UNIX son una versión compacta de estas listas. En los sistemas Windows actuales también se utilizan estos tipos de lista. Cada fichero mantiene la suya. Si se implanta por filas, cada agente mantendrá la lista de operaciones que podrá efectuar sobre cada objeto. A cada una de estas "listas" se la llama "capacidad".
C	No es un mecanismo físico. CIERTO. Es un mecanismo de seguridad, pero no de esa clase.
D	Puede utilizarse para implantar parcialmente diferentes políticas de seguridad. CIERTO. La política especifica qué hacer pero no cómo se implantará. La implantación recae en los mecanismos.
E	Todas las anteriores.
F	Ninguna de las anteriores.

15. Al considerar los riesgos…

A	Si no hay vulnerabilidades no podrá haber ataques. CIERTO. Un ataque comprende el conjunto de acciones desarrolladas durante una amenaza. Por su parte, una amenaza implica el aprovechamiento de alguna vulnerabilidad.
B	Para que no haya amenazas bastará con eliminar a todos los usuarios del sistema. FALSO. Para que no haya amenazas se tendrá que garantizar que no haya ninguna vulnerabilidad. No todas las vulnerabilidades dependen de los usuarios que pueda tener un sistema.
C	Una amenaza es la implantación de un ataque. FALSO. Un ataque es la "implantación" de una amenaza.
D	Se da una vulnerabilidad cuando un usuario malintencionado se aprovecha de una amenaza. FALSO. Se da una amenaza cuando un usuario malintencionado se aprovecha de una vulnerabilidad. Los términos clave de esa afirmación estaban intercambiados.
E	Todas las anteriores.
F	Ninguna de las anteriores. Aunque el apartado A se debe considerar verdadero, también es cierto que hay ataques "ciegos" en los que sin haber explorado previamente si el sistema objetivo tenía alguna vulnerabilidad se inicia de igual manera un ataque. Por otra parte, va a resultar imposible construir un sistema perfecto donde no haya ninguna vulnerabilidad. POR TODO LO DICHO, EN ESTA CUESTIÓN SE CONSIDERARÁN CORRECTAS TANTO LA OPCIÓN "A" COMO LA "F".

16. Sobre los tipos de amenazas en sistemas distribuidos:

A	Son internas cuando el agente que las genera tiene acceso físico a un ordenador del sistema. CIERTO. Esa es la característica que definía a una amenaza interna.
B	Son internas cuando el agente que las genere es capaz de corromper la TCB del sistema, modificando para ello los protocolos de comunicación. FALSO. Las amenazas internas guardan relación con el ordenador utilizado para generar la amenaza (debe formar parte del sistema y se necesita acceso físico a él), pero no la guardan con el objeto afectado por la amenaza (ficheros, información, recursos...)
C	Son externas cuando el agente que las provoque no es técnicamente experto. FALSO. Ésas serían las amenazas estructuradas.
D	Son externas cuando el agente que las provoque sabe cómo corromper las comunicaciones entre los agentes con acceso físico al sistema. FALSO. No es una definición válida para las amenazas externas. Tanto las internas como las externas dependen sólo de si se tiene acceso físico (interna) o no (externa) a uno de los ordenadores del sistema.
E	Todas las anteriores.
F	Ninguna de las anteriores.

17. Definiciones correctas de tipos de ataque en sistemas distribuidos:

A	Denegación de servicio: Obtención de acceso sobre un servicio violando una política activa. FALSO. La definición correcta sería "inhabilitación de servicios para los usuarios que están autorizados para utilizarlos".
B	Acceso: Descubrimiento desautorizado de servicios y vulnerabilidades. FALSO. La definición utilizada fue "obtención de acceso sobre un servicio u objeto, violando cierta política".
C	Recolección: Descubrimiento desautorizado de servicios y vulnerabilidades, utilizado posteriormente para ataques de acceso o denegación de servicio. CIERTO. Esa fue la definición dada en el tema 7.
D	Denegación de servicio: Inhabilitación de servicios para usuarios desautorizados. FALSO. Los usuarios que no estén autorizados jamás deberían acceder a un servicio. En la justificación del apartado A ya se ha dado la definición correcta.
E	Todas las anteriores.
F	Ninguna de las anteriores.

18. Ejemplos de ataques de "Recolección de información":

A	"Ping of death". FALSO. Es un ataque de tipo "denegación de servicio".
B	Congestión SYN. FALSO. Es un ataque de tipo "denegación de servicio".
C	Corrupción de paquetes. FALSO. Es un ataque de "acceso" pues implica el acceso a la información y la corrupción de ésta mientras es transmitida.
D	Denegación de servicio. FALSO. Es el nombre de otro de los tipos de ataque.
E	Todas las anteriores.
F	Ninguna de las anteriores.

19. Sobre los ataques "*man in the middle*":

A	Es un ejemplo de ataque de acceso. CIERTO. Permite acceder a la información transmitida de una forma no autorizada y tomar este hecho como base para realizar otras acciones posteriormente.
B	Se utiliza para suplantar a un agente autorizado. CIERTO. Puede utilizarse para ese fin.
C	Se puede implantar interceptando sesiones en curso. CIERTO. Se puede implantar de esa manera.
D	Permite corromper el estado del sistema, insertar nueva información o denegar el servicio a algunos agentes autorizados. CIERTO. Son tres posibles consecuencias de estos ataques.
E	Todas las anteriores.
F	Ninguna de las anteriores.

20. Sobre los protocolos criptográficos:

CUESTIÓN INVALIDADA. En las versiones en castellano y valenciano se mezcló al maquetar la versión definitiva los apartados y enunciado de dos cuestiones diferentes. El enunciado de esta cuestión debía haber sido "Sobre los códigos MAC:". En ese caso la respuesta correcta era la "C". Con la combinación utilizada, la veracidad o falsedad de cada apartado dependerá del protocolo concreto que se asuma al contestar por lo que habría más de un apartado aceptable.

A	Verifican la integridad de los mensajes.
B	Aseguran la confidencialidad de la comunicación.
C	Garantizan el no repudio.
D	Permiten ahorrar ancho de banda, pues el mensaje siempre se transmite comprimido.
E	Todas las anteriores.
F	Ninguna de las anteriores.

21. El despliegue de un servicio distribuido consiste en…

A	Instalar y configurar el software en el sistema, resolver sus dependencias y mantenerlo en funcionamiento. CIERTO. Esas son las tareas principales a desarrollar en el despliegue.
B	Garantizar la seguridad del servicio. FALSO. Aunque la seguridad siempre es un aspecto a considerar en cualquier etapa del ciclo de vida del software, no es todo lo que va a hacerse en el despliegue.
C	Diseñar y desarrollar todos sus componentes considerando su eficiencia. FALSO. Tanto el diseño como el desarrollo de los componentes deben concluir antes de iniciar el despliegue.
D	Monitorizar el uso del servicio, contabilizar sus costes y gestionar su cobro a los usuarios. FALSO. El despliegue de un servicio no consiste exclusivamente en la contabilización y gestión económica.
E	Todas las anteriores.
F	Ninguna de las anteriores.

22. El SLA es…

A	Un contrato entre el proveedor de una infraestructura y el desarrollador de aplicaciones distribuidas. FALSO. El SLA no afecta exclusivamente a los proveedores de infraestructura. No siempre será un contrato. Al menos, no puede catalogarse así en todas las legislaciones.
B	Un contrato entre el desarrollador de un servicio y sus usuarios en el que se consideran los aspectos de seguridad. FALSO. Los agentes implicados en este acuerdo no son el desarrollador y los usuarios, sino el proveedor de un servicio y sus usuarios.
C	Un acuerdo entre el proveedor de un servicio y sus usuarios en el que se consideran principalmente dos aspectos: rendimiento y disponibilidad. CIERTO. Ésta sería una definición aceptable para un SLA. Intervienen el proveedor de servicios y los clientes de ese proveedor. Dichos clientes son los usuarios del servicio. Los aspectos comúnmente considerados en ese acuerdo son la disponibilidad y el rendimiento del servicio.
D	Un compromiso entre tres propiedades de un servicio distribuido: consistencia, disponibilidad y tolerancia al particionado de la red. FALSO. Las tres propiedades mencionadas son las citadas en el teorema CAP. Ese teorema no proporciona (ni guarda apenas relación con) la definición de los acuerdos establecidos en un SLA.
E	Todas las anteriores.
F	Ninguna de las anteriores.

23. Cada instancia de un componente de un servicio distribuido…

A	Puede iniciarse y detenerse con independencia de las demás instancias. CIERTO. Si el inicio y paro de una instancia dependiera del estado de las demás (obligando a parar o arrancar todas ellas de manera simultánea), poco se ganaría al tener múltiples instancias. Habrá múltiples instancias para aumentar la capacidad de servicio e incrementar la disponibilidad. Debe existir cierta libertad para poder iniciar nuevas instancias cuando la carga soportada se vaya incrementando y para pararlas cuando la carga disminuya.
B	Puede considerarse una réplica del componente. CIERTO. La justificación del apartado anterior así lo sugiere. Cada instancia aporta capacidad de servicio adicional. También incrementa la disponibilidad, tolerándose el fallo de algunas de ellas.
C	Debería desplegarse de tal manera que su probabilidad de fallo sea independiente de la probabilidad de las demás instancias. CIERTO. Cada instancia debe ubicarse en un nodo distinto y cada nodo deberá depender de diferentes fuentes posibles de fallo: deberían estar en centros de datos diferentes (o al menos en "racks" diferentes en caso de ubicarse en un mismo centro), tener diferentes puntos de acceso a la red de comunicaciones... Así, en caso de que haya algún problema físico (corte de alimentación eléctrica en el centro, inundaciones, avería en la red...) sólo afectará a una o unas pocas instancias. Las demás superarán esa situación de fallo.
D	Se ubica con independencia de la ubicación de las instancias de otros servicios. CIERTO. También es consecuencia de lo dicho en los apartados anteriores. Hay que reducir las dependencias sobre cualquier otro elemento del sistema.
E	Todas las anteriores.
F	Ninguna de las anteriores.

24. En la gestión del ciclo de vida de un servicio hay que considerar...

A	Cuántos programadores participan en el desarrollo de los componentes. FALSO. Esos detalles de la etapa de desarrollo no afectan a la gestión del ciclo de vida del servicio. Esta gestión afecta al SERVICIO no a los programas utilizados en los servidores. Un servicio podrá iniciarse, pararse, reanudarse, sufrir actualizaciones y variar el número de instancias que dan soporte a cada componente, en función de la carga que deba soportarse. Todas esas acciones son las relevantes en el ciclo de vida del servicio.
B	Cómo actualizar los componentes y cuándo y cómo habrá que añadir o eliminar réplicas de cada componente. CIERTO. Ésas son algunas de las acciones a tener en cuenta.
C	Los protocolos de enlace y de red utilizados para intercomunicar los componentes. FALSO. Esos protocolos son responsabilidad del subsistema de comunicaciones, en sus niveles 2 y 3, respectivamente. Normalmente los servicios se implantarán a nivel de aplicación dentro de la arquitectura de niveles del sistema de comunicaciones.
D	Si los componentes han sido implantados bajo un modelo multi-hilo o un modelo de programación asincrónica. FALSO. Eso no tiene por qué afectar a las acciones importantes dentro del ciclo de vida del servicio (enumeradas en la justificación del primer apartado de esta cuestión).
E	Todas las anteriores.
F	Ninguna de las anteriores.

25. Aspectos que deben decidirse durante el despliegue de un servicio distribuido:

A	De qué otros servicios depende y qué SLA se requiere de cada uno de ellos.
B	El orden de inicio de los componentes del servicio.
C	Cuántas instancias tendrá cada componente.
D	En qué nodos se instalará y ejecutará cada instancia.
E	Todas las anteriores. Todos los aspectos citados en los apartados anteriores deben ser considerados a la hora de desplegar un servicio, por sencillo que éste llegue a ser.
F	Ninguna de las anteriores.

26. Un descriptor de despliegue incluye…

A	Las garantías de seguridad del servicio a desplegar. FALSO. Las garantías de seguridad no suelen especificarse en el descriptor de despliegue.
B	El SLA ofrecido a los futuros usuarios del servicio a desplegar. FALSO. El despliegue se realizará teniendo en cuenta el SLA, pero el SLA no está incluido en el descriptor de despliegue.
C	La configuración de cada una de las instancias a desplegar, el nodo donde ubicar cada una y una descripción de las dependencias internas y externas a resolver. CIERTO. Ése es el conjunto de información incluido en el descriptor de despliegue.
D	El código de cada uno de los componentes del servicio. FALSO. El descriptor contiene la configuración de las diferentes instancias pero generalmente no incluye el código de cada componente.
E	Todas las anteriores.
F	Ninguna de las anteriores.

27. Los componentes de un servicio necesitan actualizarse para…

A	Eliminar vulnerabilidades.
B	Mejorar su eficiencia.
C	Eliminar errores de programación.
D	Ampliar su funcionalidad y adaptarse a nuevas configuraciones.
E	Todas las anteriores. Todos los apartados anteriores son ejemplos válidos de razones que conducen a actualizar un componente.
F	Ninguna de las anteriores.

28. El despliegue en un sistema IaaS:

A	Está automatizado. Basta con rellenar las plantillas de despliegue y el sistema se encarga de todo. FALSO. Llegará a automatizarse en los futuros sistemas PaaS, pero no está automatizado en ningún sistema IaaS.
B	Es responsabilidad del proveedor de infraestructura. FALSO. Si fuera responsabilidad del proveedor, los usuarios de un sistema IaaS obtendrían la imagen de que el despliegue está automatizado para ellos. No es así.
C	Es responsabilidad del administrador del sistema IaaS. FALSO. En un sistema IaaS el rol de administrador no está claramente delimitado. Algunas tareas de administración las realiza el proveedor y otras las realiza el cliente de este tipo de servicios.
D	El proveedor IaaS proporciona el número solicitado de máquinas virtuales y el desarrollador del servicio se encarga de su despliegue. CIERTO. Al estar repartido de esta manera resulta imposible automatizar el despliegue de un servicio distribuido en un sistema IaaS.
E	Todas las anteriores.
F	Ninguna de las anteriores.

29. En un sistema PaaS, el despliegue se supone que...

A	Está automatizado. Basta con rellenar las plantillas de despliegue y el sistema se encarga de todo. CIERTO. Ésta es una de sus principales ventajas frente a los sistemas IaaS.
B	No es responsabilidad del PaaS. Es responsabilidad del proveedor SaaS. FALSO. Sí que es responsabilidad del PaaS. Es la diferencia entre una plataforma y una infraestructura. La plataforma automatiza el despliegue.
C	No es responsabilidad del PaaS. Es responsabilidad del IaaS subyacente. FALSO. Sí que es responsabilidad del PaaS. Es la diferencia entre una plataforma y una infraestructura. La plataforma automatiza el despliegue.
D	El proveedor PaaS proporciona el número solicitado de máquinas virtuales y el desarrollador del servicio se encarga de su despliegue. FALSO. El proveedor de una plataforma no se encarga de la gestión de máquinas virtuales. Eso es responsabilidad del proveedor de la infraestructura.
E	Todas las anteriores.
F	Ninguna de las anteriores.

30. Si se automatizara la gestión del ciclo de vida de un servicio distribuido…

A	El servicio se adaptaría sin problemas a los cambios en la carga soportada, consumiendo un volumen óptimo de recursos, reduciendo su coste.
B	El servicio podría considerarse elástico.
C	Se garantizaría (bajo los límites del SLA acordado) la disponibilidad del servicio.
D	Se ofrecería un rendimiento y funcionalidad acordes con su SLA.
E	Todas las anteriores. Todo lo que se comenta en los apartados anteriores es consecuencia de la automatización en la gestión del ciclo de vida de un servicio.
F	Ninguna de las anteriores.

31. Los patrones arquitectónicos básicos…

A	Describen cuántos componentes puede tener un servicio. FALSO. Los componentes que deberá tener un servicio se recogerán en su diseño. Un patrón arquitectónico especifica de qué manera interactuarán dos componentes.
B	Proporcionan una guía completa para decidir cuántas instancias de cada componente habrá que desplegar. FALSO. El número de instancias que deberán desplegarse dependerá de la carga soportada en cada momento y de la calidad de servicio comprometida en el SLA. Eso no se recoge en un patrón arquitectónico.
C	Describen los patrones básicos de comunicación. CIERTO. Ese es el objetivo de un patrón arquitectónico.
D	Proporcionan el SLA más sencillo posible para cada servicio. FALSO. Un patrón arquitectónico no contempla los acuerdos que puedan establecer el proveedor y el cliente de un servicio.
E	Todas las anteriores.
F	Ninguna de las anteriores.

32. El patrón petición/respuesta...

A	Intercomunica a dos agentes en su caso básico.
B	Es un patrón doblemente sincrónico.
C	Evita la concurrencia en el agente cliente.
D	Un defecto en el servidor bloqueará al cliente si este último ya había enviado su petición.
E	Todas las anteriores. Los aspectos mencionados en estos apartados caracterizan a este patrón
F	Ninguna de las anteriores.

33. Semánticas en caso de reinicio del servidor en el patrón petición/respuesta...

A	La semántica "al menos una vez" sería recomendable en caso de utilizar operaciones idempotentes. CIERTO. Las operaciones idempotentes siempre generan el mismo resultado, independientemente del número de veces que lleguen a ejecutarse. Por tanto, conviene que estas operaciones se ejecuten al menos una vez. Si se ejecutasen más de una no se generará ningún mal comportamiento ni se generará ninguna inconsistencia en el estado de los servidores.
B	La semántica "al menos una vez" es la que debe utilizarse en caso de que también pueda fallar el cliente. FALSO. Si el servidor no pudiera gestionar adecuadamente las repeticiones de las peticiones realizadas por los clientes, esta semántica podría llegar a generar inconsistencias en el estado. Por tanto, depende de cómo se comporte el servidor ante mensajes repetidos. Que falle o no el cliente y en qué momento llegue a fallar no siempre condiciona la semántica asumida. De hecho, si el cliente fallara podría llegarse a perder la última petición que pretendía iniciar mientras falló. En ese caso habría resultado más fácil seguir una semántica "como máximo una vez".
C	La semántica "como máximo una vez" realiza al menos un reenvío de la petición. FALSO. Los reenvíos se utilizarán en la semántica "al menos una vez". Son innecesarios en la semántica "como máximo una vez".
D	La semántica "como máximo una vez" es la que debe utilizarse cuando no se replique el servidor. FALSO. Tanto una semántica como otra dependerán del tipo de operación a ejecutar. El hecho de que el servidor esté replicado no condiciona el uso de una semántica u otra.
E	Procesador, Causal. Esta opción estaba originalmente mal (son modelos de consistencia del parcial anterior) y se modificó en el examen pasando a ser "Todas las anteriores".
F	Procesador, Causal, Caché, FIFO. Esta opción estaba originalmente mal (son modelos de consistencia del parcial anterior) y se modificó en el examen pasando a ser "Ninguna de las anteriores".

34. El patrón PUSH-PULL...

A	Es un patrón doblemente sincrónico. FALSO. Es un patrón asincrónico. No exige el bloqueo del emisor ni del receptor.
B	Es un patrón de comunicación bidireccional. FALSO. Es un patrón unidireccional. Los mensajes siempre parten del socket PUSH y van a parar al socket PULL.
C	Asume consistencia causal. FALSO. No asume ni exige ningún tipo de consistencia predeterminado.
D	Es un patrón de comunicación unidireccional. CIERTO.
E	Todas las anteriores.
F	Ninguna de las anteriores.

35. El patrón PUSH-PULL...

A	Limita la escalabilidad al introducir bloqueos prolongados en caso de relacionar en una cadena a múltiples componentes. FALSO. No introduce bloqueos, por ser asincrónico.
B	Es un patrón asincrónico. CIERTO. Ya se ha comentado en el apartado A de la cuestión anterior.
C	Asume consistencia secuencial. FALSO. No asume ni exige ningún modelo de consistencia predeterminado.
D	Exige una semántica "al menos una vez". FALSO. No exige ninguna semántica de entrega de operaciones. Tampoco está claro que los componentes a intercomunicar sigan siempre un modelo cliente/servidor. Para eso ya existe un patrón petición/respuesta.
E	Todas las anteriores.
F	Ninguna de las anteriores.

36. Al desplegar un patrón PUSH-PULL en ZMQ...

A	Los sockets PUSH se configurarán con las direcciones de los sockets PULL. FALSO. Depende de cuál sea el componente estable.
B	Los sockets PULL se configurarán con las direcciones de los sockets PUSH. FALSO. Depende de cuál sea el componente estable.
C	Tanto los sockets PUSH como los PULL se configurarán con las direcciones de los sockets del otro tipo. FALSO. Depende de cuál sea el componente estable. Es difícil que ambos componentes sean estables y necesiten una configuración como la que se comenta en este apartado.
D	Los sockets de tipo menos estable se configurarán con la dirección del socket de tipo más estable. Qué tipo será más estable dependerá del servicio a desplegar. CIERTO. Esta fue la regla a seguir explicada en las clases.
E	Todas las anteriores.
F	Ninguna de las anteriores.

37. El patrón PUB-SUB...

A	Es un patrón de comunicación bidireccional. FALSO. En un patrón de comunicación unidireccional. Los mensajes son siempre emitidos por el socket PUB y entregados en los sockets SUB.
B	Utiliza comunicación sincrónica. FALSO. Utiliza comunicación asincrónica.
C	Al desplegarlo se suelen configurar los sockets PUB con las direcciones de los sockets SUB. FALSO. Por ser un patrón diseñado para realizar difusiones, es más lógico que el componente estable sea el emisor y que sea responsabilidad de los suscriptores/receptores el configurarse con la dirección de ese componente estable. Si se hiciera al contrario, el publicador debería reconfigurarse cada vez que se añadiera un nuevo suscriptor.
D	Se utiliza para difundir mensajes desde el socket SUB a los sockets PUB. FALSO. Como ya se ha explicado en el primer apartado, la comunicación unidireccional sigue el sentido contrario. Desde PUB hacia SUB.
E	Todas las anteriores.
F	Ninguna de las anteriores.

.

38. Una arquitectura petición/respuesta avanzada puede implantarse utilizando una cola de comunicación intermedia (o agente "bróker"). En esa arquitectura...

A	El "bróker" suele utilizar dos sockets ROUTER.
B	Los servidores (o trabajadores) pueden sustituir sus sockets REP por sockets REQ.
C	Los servidores (o trabajadores) pueden sustituir sus sockets REP por sockets DEALER.
D	Tendrá una recuperación delicada cuando falle el "bróker", pues suele ser el componente estable en este patrón arquitectónico.
E	Todas las anteriores. Como se ha podido ver en prácticas y en el tema 9 de teoría, todos los apartados anteriores son ciertos.
F	Ninguna de las anteriores.

39. En una arquitectura cliente/servidor avanzada puede necesitarse un mecanismo para detectar y rechazar peticiones duplicadas si…

CUESTIÓN ANULADA. En algunos grupos de teoría no se explicó la semántica "exactamente una vez". La respuesta correcta era la "D" y podía deducirse de la hoja de la presentación del tema 9 en la que se explicaba ese mecanismo de detección y descarte de peticiones duplicadas.

A	Se utiliza una semántica "al menos una vez" y la petición es idempotente.
B	Se utiliza una semántica "al menos una vez" y el servidor no está replicado.
C	Se utiliza una semántica "exactamente una vez" y la petición es idempotente.
D	Se utiliza una semántica "exactamente una vez" y la petición no es idempotente.
E	Todas las anteriores.
F	Ninguna de las anteriores.

40. Algunos problemas de la arquitectura PUB/SUB son...

A	Pérdida de mensajes en suscriptores lentos o con inicio tardío en su suscripción. CIERTO. Estos son los dos problemas principales de este patrón arquitectónico. En el tema 9 se describieron algunas soluciones para cada problema.
B	Despliegue difícil. No resulta fácil decidir qué agentes pueden considerarse estables. FALSO. El componente estable siempre será el publicador/difusor.
C	Utiliza un patrón de comunicación bidireccional sincrónico que difícilmente puede escalar. FALSO. Utiliza un patrón unidireccional asincrónico fácilmente escalable.
D	Debe usar un agente intermediario que podrá fallar. FALSO. El agente intermediario o "bróker" se podría necesitar en un patrón petición/respuesta avanzado pero no se ha dicho en ningún momento que pueda necesitarse para un patrón publicación/suscripción.
E	Todas las anteriores.
F	Ninguna de las anteriores.