## RECOPILACIÓN PREGUNTAS UNIDAD 7 (desde 2014)

**[Examen FINAL 2º bloque junio 2020 --> examen online]**

**BATERIA Tema 7 - Definición Sistema Distribuido y middleware + Disponibilidad y fallos => 16 preg** => seleccionar 8

1. Todo sistema concurrente es un sistema distribuido. \*Falso
2. Todo sistema distribuido está formado por varias máquinas autónomas. \*Verdadero
3. Los nodos de un sistema distribuido no comparten memoria, pero comparten un reloj global.\*Falso
4. Si necesitamos un sistema escalable y con alta disponibilidad la única opción es un Sistema Distribuido. \*Verdadero
5. Todo Sistema Distribuido proporciona paralelismo real. \*Verdadero
6. Un diseño concurrente dado se implanta de forma más simple en un Sistema Distribuido que en un sistema centralizado. \*Falso
7. Para conseguir transparencia de fallos necesitamos replicación. \*Verdadero
8. La transparencia de concurrencia implica que el sistema distribuido sigue funcionando sea o no un sistema concurrente. \*Falso
9. La transparencia únicamente puede cumplirse (transparencia total) o no cumplirse (no hay transparencia). \*Falso
10. La disponibilidad garantiza que el servicio está disponible tanto para usuarios autorizados como los que no lo están. \*Falso
11. La capa de middleware integra mecanismos que facilitan la programación de aplicaciones distribuidas. \*Verdadero
12. Un sistema distribuido proporcionará diferentes tipos de transparencia. Entre ellos: transparencia en el rendimiento, en la escalabilidad, en su disponibilidad, en la seguridad... \*Falso
13. Un Sistema Distribuido ofrece al exterior una imagen de máquina única.

\*Verdadero

**BATERIA Tema 7 - Disponibilidad y fallos + Escalabilidad y Replicación activa/pasiva => 20 preg** => seleccionar 8

1. Todo fallo compuesto es detectable. \*Falso
2. La partición de red es un fallo compuesto. \*Verdadero
3. El fallo de parada es un fallo simple detectable. \*Verdadero
4. Todo fallo simple detectable se puede convertir a fallo de parada. \*Verdadero
5. El único tipo de fallo simple detectable es el de fallo-parada (el nodo deja de enviar mensajes). \*Falso
6. Una disponibilidad del 99,99% indica que el sistema, a lo largo de 10.000 días, podrá no estar disponible en total durante un día o menos. \*Verdadero
7. Un ejemplo de fallo indetectable es que un sistema tarde demasiado tiempo en responder. \*Falso
8. El módulo de detección de fallos avisa al servicio de pertenencia a grupo cuando sospecha que un nodo está fallando. \*Verdadero
9. Para que un sistema distribuido se considere bien diseñado, ha de ofrecer simultáneamente consistencia fuerte, alta disponibilidad y permitir que en caso de particiones, todas ellas sigan funcionando. \*Falso
10. Los fallos de parada no pueden detectarse directamente por otros nodos. Se requiere de algoritmos de consenso o de quórum para poder tratar este tipo de fallos. \*Falso
11. La replicación activa no necesita reconfiguración tras el fallo. \*Verdadero
12. La replicación pasiva no necesita reconfiguración tras el fallo de cualquiera de sus réplicas. \*Falso
13. Para mejorar la escalabilidad es conveniente repartir el procesamiento entre los distintos nodos.\*Verdadero
14. Cuando un sistema distribuido permite ampliar y reducir sus recursos para acomodar cargas más pesadas o ligeras, según se requiera, decimos que es escalable. \*Verdadero
15. Para construir un sistema escalable es necesario considerar el número de usuarios que lo utilizará, pero no es necesario tener en cuenta cuántos recursos gestionará dicho sistema. \*Falso
16. Gracias a la técnica del caching es posible conseguir consistencia fuerte. \*Falso
17. En la replicación pasiva todas las réplicas secundarias se actualizan en el mismo instante. \*Falso
18. En la replicación activa es necesario emplear algoritmos de difusión que proporcionen la misma secuencia de mensajes a todas las réplicas. \*Verdadero
19. La replicación es una de las técnicas que contribuye a conseguir sistemas con alta disponibilidad. \*Verdadero
20. La forma habitual de tratar una partición consiste en utilizar algoritmos que permitan que las réplicas mantengan siempre entre sí una consistencia fuerte. \*Falso

# [2º Parcial junio 2020 --> examen online]

## BATERIA Tema 7 - Definición Sistema Distribuido y middleware + Transparencia y seguridad => batería 24 preg -> Sel. 9

1. Un Sistema Distribuido siempre será concurrente, pues habrá múltiples actividades en el sistema que cooperarán entre sí. \*Verdadero
2. Un Sistema Distribuido siempre exigirá una planificación expulsiva por prioridades fijas, pues será de tiempo real. \*Falso
3. Todo sistema en red es un sistema distribuido. \*Falso
4. Un Sistema Distribuido siempre se implanta sobre un conjunto de nodos interconectados en red. \*Verdadero
5. Un middleware es parte integral del sistema operativo. \*Falso
6. La capa de middleware determina el modelo de programación a utilizar para el desarrollo de las aplicaciones (objetos distribuidos, dirigida por eventos, etc.).

\*Verdadero

1. El middleware utiliza los servicios de las aplicaciones distribuidas. \*Falso
2. El middleware orientado a mensajería solo proporciona primitivas de envío y recepción para comunicación directa entre procesos. \*Falso
3. La capa de middleware se ubica bajo el nivel de aplicación. \*Verdadero
4. La transparencia de distribución proporciona la imagen de sistema único.

\*Verdadero

1. La transparencia de ubicación implica que no necesitamos saber en qué nodo está cada recurso. \*Verdadero
2. El que un servicio de máquinas virtuales pueda reactivar una máquina virtual cuyo estado se había guardado previamente en almacenamiento secundario si detecta que algún usuario quiere volver a utilizarla es un ejemplo de transparencia de persistencia.

\*Verdadero

1. El que Excel Online permita que varios usuarios utilicen simultáneamente una hoja de cálculo es un ejemplo de transparencia de replicación. \*Falso
2. La transparencia de persistencia oculta el hecho de que un recurso esté ubicado en memoria volátil o en memoria persistente..\*Verdadero
3. Cuando un sistema distribuido permite acceder a los recursos sin conocer el dispositivo exacto en el que se encuentran, proporciona transparencia de ubicación.

\*Verdadero

1. La transparencia de ubicación, la transparencia de replicación y la transparencia de fallos representan los principales ejes de la transparencia de distribución.

\*Verdadero

1. Los detectores de fallos y la replicación son mecanismos para lograr transparencia de fallos. \*Verdadero
2. La autenticación garantiza que únicamente tienen acceso a los datos los usuarios autorizados. \*Falso
3. La integridad es una característica de seguridad que indica que sólo los usuarios autenticados y autorizados pueden acceder a determinados recursos. \*Falso
4. RMI proporciona transparencia de ubicación. \*Verdadero
5. Para conseguir transparencia de fallos necesitamos replicación. \*Verdadero
6. Los servicios de nombres ayudan a proporcionar transparencia de ubicación.

\*Verdadero

1. En el acceso por parte de clientes remotos debemos evitar la suplantación de identidad. \*Verdadero
2. La autenticación es una característica común de seguridad que también está presente en los sistemas distribuidos. \*Verdadero

## BATERIA Tema 7 - Disponibilidad y fallos + Escalabilidad y Replicación activa/pasiva. 24 preguntas -> Sel 9

1. La consistencia representa la coherencia entre las réplicas de un mismo recurso.

\*Verdadero

1. La disponibilidad representa la probabilidad de que al acceder al sistema el servicio esté disponible. \*Verdadero
2. La consistencia fuerte es imposible. \*Falso
3. Todo fallo compuesto es indetectable. \*Falso
4. La pertenencia a grupo implica consenso para decidir si una sospecha de fallo se considera fallo. \*Verdadero
5. Los fallos bizantinos son fallos simples no detectables. \*Verdadero
6. El teorema CAP indica que en un sistema hay que elegir entre consistencia (C), disponibilidad (A) y tolerar particiones (P) (elegir una de las tres). \*Falso
7. Para resolver un fallo bizantino se recurre al módulo de detección de fallos. \*Falso
8. La forma habitual de tratar una partición consiste en utilizar algoritmos que permitan que las réplicas puedan converger cuando finalice la causa por la que se había producido la partición. \*Verdadero
9. Una elevada disponibilidad del sistema implica que aunque pongamos en marcha tareas de mantenimiento, los usuarios puedan seguir accediendo a los servicios del sistema. \*Verdadero
10. En los fallos bizantinos los nodos proporcionan una respuesta errónea no detectable, por ejemplo debida a errores en el hardware o errores en el software.

\*Verdadero

1. En replicación activa todas las copias reciben y procesan la misma secuencia de operaciones. \*Verdadero
2. En la replicación pasiva es necesario emplear algoritmos de difusión que proporcionen la misma secuencia de mensajes a todas las réplicas. \*Falso
3. La replicación pasiva permite detectar fallos bizantinos. \*Falso
4. En replicación pasiva las copias secundarias reciben mensajes de actualización.

\*Verdadero

1. Para mejorar la escalabilidad es conveniente centralizar los datos. \*Falso
2. Un sistema distribuido se caracteriza por ofrecer escalabilidad, o lo que es lo mismo, consigue que los servicios que ofrece están siempre disponibles. \*Falso
3. En la replicación activa se acuerda, para cada petición de un cliente, qué réplica atenderá la petición. \*Falso
4. La distribución de datos utilizada para conseguir sistemas escalables consiste en hacer que cada una de las réplicas mantenga una copia de todos los datos del sistema distribuido. \*Falso
5. La replicación es una de las técnicas que pueden contribuir a conseguir sistemas escalables. \*Verdadero
6. Para tratar fallos simples indetectables debemos utilizar replicación activa.

\*Verdadero

1. La replicación activa no admite operaciones no deterministas. \*Verdadero
2. En la replicación pasiva la réplica primaria recibe todas las peticiones de los clientes. \*Verdadero
3. La técnica de *caching* es un caso particular de la replicación donde el cliente mantiene una réplica de los datos que mantiene el servidor, con consistencia débil.

\*Verdadero

# [FINAL 2019]

Sobre la transparencia en los sistemas distribuidos:

|  |  |
| --- | --- |
| 1. Los detectores de fallos y la replicación son mecanismos para lograr transparencia de fallos. | V |
| 2. La transparencia de concurrencia persigue ocultar la ubicación de los recursos. | F |
| 3. La transparencia de persistencia trata de ocultar el hecho de que los recursos  estén almacenados de forma no volátil. | V |
| 4. La transparencia en sistemas distribuidos suele implicar coste para lograrla y suele implicar mayor calidad observada por los usuarios, respecto a no ofrecer tal transparencia. | V |
| 5. Los principales ejes de la transparencia de distribución son tres: transparencia de  ubicación, transparencia de fallos y transparencia de replicación. | V |

Respecto a la disponibilidad en los sistemas distribuidos:

|  |  |
| --- | --- |
| 6. Los únicos factores que afectan a la disponibilidad son los fallos y las tareas de mantenimiento. | F |
| 7. El principal mecanismo para lograr tolerancia a fallos es la replicación. | V |
| 8. Los fallos bizantinos son fallos compuestos. | F |
| 9. Las particiones son fallos detectables, pues los nodos pueden saber en todo momento si el sistema está particionado en dos o más particiones en ejecución. | F |

Sobre los modelos de replicación:

|  |  |
| --- | --- |
| 10. En la replicación pasiva, la réplica primaria envía mensajes de  actualización de estado (*checkpoint*) a las otras réplicas. | V |
| 11. La replicación activa requiere de menor trabajo de reconfiguración en caso de fallos que la replicación pasiva. | V |
| 12. La replicación pasiva suele ser más eficiente que la replicación activa durante la operativa del sistema en ausencia de fallos. | V |
| 13. La replicación activa requiere que las réplicas se ejecuten de acuerdo a | V |

|  |  |
| --- | --- |
| un modelo de ejecución determinista. |  |
| 14. La replicación activa requiere del empleo de algoritmos de difusión ordenada de mensajes. | V |

Sobre la escalabilidad:

|  |  |
| --- | --- |
| 15. Entre las técnicas más importantes para lograr escalabilidad, podemos mencionar replicación y distribución de la carga entre diferentes  nodos. | V |
| 16. Un sistema que emplee caching generalmente será menos escalable que otro sistema que no emplee dicha técnica. | F |
| 17. Los sistemas altamente escalables suelen garantizar consistencia fuerte. | F |
| 18. La técnica de distribuir datos mejora la escalabilidad. | V |

## [2º parcial 2019]

Sobre los sistemas distribuidos:

|  |  |
| --- | --- |
| 1. Un sistema distribuido es un conjunto de ordenadores independientes que ofrecen a sus usuarios la imagen de un sistema coherente único. | V |
| 2. De forma general, los fallos compuestos se tratan de igual forma a la aparición de varios fallos simples de forma consecutiva. | V |
| 3. En un sistema distribuido, a nivel hardware las máquinas comparten entre sí los recursos (memoria, reloj, disco, etc...). | F |
| 4. Los fallos, las tareas de mantenimiento y los ataques maliciosos son tres  factores que afectan a la escalabilidad del sistema. | F |
| 5. Para lograr transparencia de ubicación, los recursos requieren estar  identificados con nombres simbólicos únicos. | V |

Sobre la escalabilidad y disponibilidad de los sistemas distribuidos:

|  |  |
| --- | --- |
| 6. En general, la técnica de replicación permite aumentar tanto la escalabilidad del sistema como su disponibilidad. | V |
| 7. La técnica de *caching* es un caso particular de la replicación donde el cliente mantiene una réplica exacta, con consistencia fuerte, de los datos  que mantiene el servidor. | F |
| 8. La distribución del procesamiento entre diferentes nodos y el particionado de los datos permite aumentar la escalabilidad del sistema. | V |
| 9. El teorema CAP nos indica que, en un sistema a gran escala, donde las particiones ocurren, se debe sacrificar disponibilidad del sistema, o bien su consistencia. | V |
| 10. En la replicación activa, en caso de fallo de una réplica, el trabajo de  reconfiguración consiste en que la réplica primaria enviará un mensaje menos de *checkpoint*. | F |

Sobre la escalabilidad y disponibilidad de los sistemas distribuidos:

|  |  |
| --- | --- |
| 1. El servicio de pertenencia a grupo permite determinar quiénes son los nodos que están activos, y utiliza para ello el algoritmo de elección de | F |



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

|  |  |
| --- | --- |
| líder en anillo. |  |
| 2. Los detectores de fallos, el servicio de pertenencia a grupo y la replicación son mecanismos que permiten aumentar la disponibilidad y lograr tolerancia a fallos. | V |
| 3. Se produce una partición en un sistema distribuido cuando se producen varios fallos en nodos o canales de comunicación que dejan al sistema  dividido en dos o más subgrupos. | V |
| 4. Tenemos un sistema con varios nodos que miden la temperatura ambiente y se la transmiten a un nodo coordinador. El coordinador puede determinar como fallo simple detectable la situación en la que un  nodo tarda mucho tiempo en contestar. | V |
| 5. Tenemos un sistema con varios nodos que miden la temperatura ambiente y se la transmiten a un nodo coordinador. El coordinador puede utilizar la replicación (con varios nodos midiendo la temperatura  de la misma zona) para poder detectar los fallos bizantinos. | V |
| 6. Los fallos de parada no pueden detectarse directamente por otros nodos.  Se requiere de algoritmos de consenso o de quórum para poder tratar este tipo de fallos. | F |

# [FINAL 2018]

Sobre la definición de Sistemas Distribuidos y sus características intrínsecas:

|  |  |
| --- | --- |
| 1. Todo sistema concurrente es un sistema distribuido | F |
| 2. Todo sistema distribuido se implanta sobre un conjunto de nodos interconectados en red. | V |
| 3. Todos los sistemas de tiempo real son distribuidos. | F |
| 4. Whatsapp es un ejemplo de sistema distribuido | V |
| 5. Los sistemas distribuidos tratan de proporcionar la imagen de sistema único, es decir que el usuario o cliente del sistema lo observe como si no estuviera distribuido. | V |

Sobre la escalabilidad y la seguridad en sistemas distribuidos:

|  |  |
| --- | --- |
| 6. Para mejorar la escalabilidad debemos distribuir los datos, pero centralizar las responsabilidades | F |
| 7. El empleo de caching no mejora la escalabilidad de los sistemas, tan sólo mejora en algunos casos el rendimiento. | F |

Sobre la transparencia en sistemas distribuidos:

|  |  |
| --- | --- |
| 8. Suele implicar un cierto sobrecoste respecto a no proporcionar tal transparencia. | V |
| 9. En muchos casos es imposible lograr una transparencia al 100%. | V |
| 10. Existen múltiples ejes de transparencia. Algunos de ellas son: independencia, acceso remoto,  ubicuidad, eficiencia. | F |
| 11. Cuando un sistema distribuido oculta en la identificación de los recursos el lugar donde se ubican, decimos que proporciona transparencia de identificación. | F |
| 12. Cuando un sistema distribuido ofrece transparencia de fallos, los usuarios perciben claramente si un componente del sistema falla. | F |

Sobre la estandarización y los principios de diseño:

|  |  |
| --- | --- |
| 13. Modularidad significa descomponer el diseño en componentes más simples | V |
| 14. La estandarización facilita la utilización de componentes (módulos) desarrollados por otros | V |

Sobre el concepto de middleware:

|  |  |
| --- | --- |
| 15. El middleware es una capa software que facilita el desarrollo de aplicaciones distribuidas | V |

|  |  |
| --- | --- |
| 16. El middleware condiciona la arquitectura software a utilizar en el desarrollo de las aplicaciones distribuidas | V |

# [2º PARCIAL 2018]

Sobre replicación en sistemas distribuidos:

|  |  |
| --- | --- |
| 1. Replicación pasiva implica que tendremos consistencia débil, pues las réplicas secundarias siempre tendrán un estado muy diferente al estado de la réplica primaria. | F |
| 2. Si se dan las premisas para poder replicar un servicio mediante replicación activa y disponemos de un protocolo de difusión de mensajes que garantice la entrega de los mensajes a las diferentes réplicas en el mismo orden, podremos lograr un sistema con  consistencia fuerte. | V |
| 3. Mediante replicación activa, si pretendemos tolerar fallos bizantinos, sólo se puede disponer de un número impar de réplicas. | F |

Sobre las características de los sistemas distribuidos:

|  |  |
| --- | --- |
| 4. La transparencia de distribución posee distintos ejes: de fallos, de ubicación, de replicación, etc. | V |
| 5. Todos los sistemas distribuidos son concurrentes. | V |
| 6. La estandarización facilita la interoperabilidad y la portabilidad de las aplicaciones. | V |
| 7. El middleware reemplaza al Sistema Operativo Local. | F |
| 8. Transparencia de fallos significa que en caso de fallos hay que notificar al usuario del fallo,  para que el usuario realice alguna acción. | F |
| 9. Uno de los mecanismos habituales para lograr transparencia de ubicación consiste en el empleo de servicios de nombres. | V |

Sobre la disponibilidad, la escalabilidad y la seguridad de los sistemas distribuidos:

|  |  |
| --- | --- |
| 10. Los fallos de temporización son indetectables. | F |
| 11. Los fallos bizantinos son fallos compuestos. | F |
| 12. El teorema CAP indica que en presencia de particiones no podemos conseguir de forma  simultánea elevada disponibilidad y un modelo de consistencia fuerte. | V |
| 13. Para tratar fallos simples indetectables debemos utilizar replicación pasiva. | F |
| 14. El uso de cachés es una de la técnicas que mejoran la escalabilidad. | V |
| 15. Integridad de los datos significa proporcionar la garantía de que el servicio no se interrumpe. | F |

# [FINAL 2017]

Sobre los sistemas distribuidos:

|  |  |
| --- | --- |
| 17. Todos los sistemas distribuidos de tiempo real son concurrentes. | V |
| 18. Todos los sistemas concurrentes son distribuidos. | F |
| 19. Whatsapp es un ejemplo de sistema distribuido. | V |
| 20. Los sistemas distribuidos tratan de proporcionar la imagen de sistema único, es decir que el usuario o cliente del sistema lo observe como si no estuviera distribuido. | V |

Sobre la transparencia en sistemas distribuidos:

|  |  |
| --- | --- |
| 21. Existen múltiples clases de transparencia. Algunas de ellas son: ubicación,  migración, replicación, fallos. | V |
| 22. Algunas veces es imposible lograr la transparencia al 100%. | V |
| 23. Los servicios de nombres ayudan a proporcionar transparencia de ubicación. | V |

# [2º PARCIAL 2017]

Sobre las características de los sistemas distribuidos:

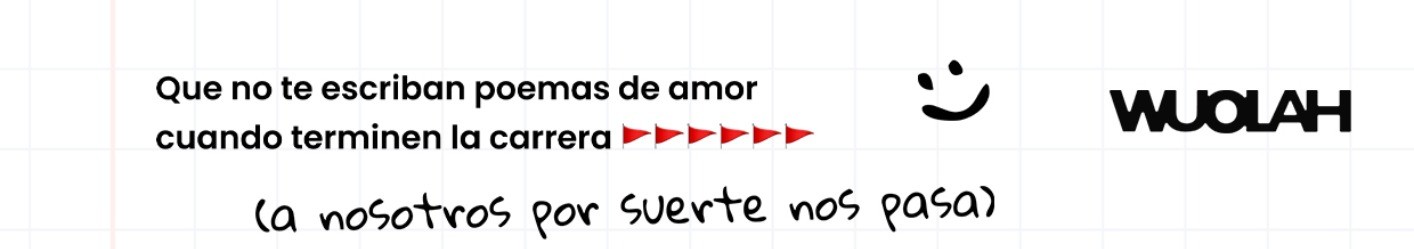
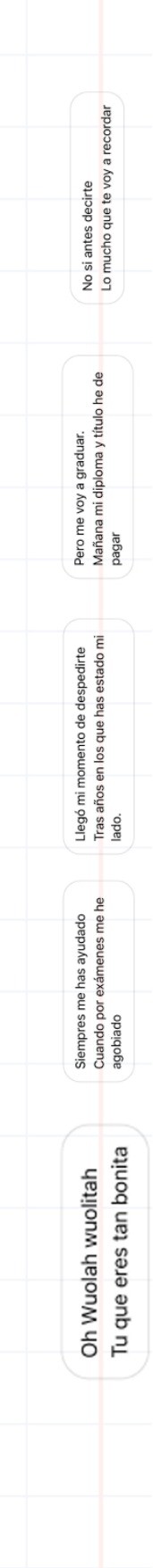
|  |  |
| --- | --- |
| 24. Cuando un sistema distribuido ofrece transparencia de fallos, los usuarios perciben claramente si un componente del sistema falla. | F |
| 25. Las características de los sistemas distribuidos son acceso homogéneo,  disponibilidad, ocultación y escalabilidad. | V |
| 26. Cuando un sistema distribuido oculta en la identificación de los recursos el lugar  donde se ubican, decimos que proporciona transparencia de identificación. | F |
| 27. El uso de estándares permite la construcción de sistemas distribuidos formados por módulos, incluso de distintos fabricantes, fácilmente reemplazables o ampliables. | V |
| 28. La escalabilidad administrativa se consigue utilizando técnicas de replicación. | F |
| 29. Para conseguir escalabilidad de distancia es necesario tener en cuenta que al aumentar la distancia entre nodos, los tiempos de respuesta y la fiabilidad  empeoran. | V |
| 30. En general, la escalabilidad se ve amenazada cuando se adoptan estrategias  descentralizadas. | F |

# [FINAL 2016]

Sobre las características de los sistemas distribuidos:

|  |  |  |
| --- | --- | --- |
| 1. | En el diseño e implementación de sistemas distribuidos se deben tener en cuenta cuatro objetivos fundamentales: facilitar acceso a los recursos remotos, proporcionar transparencia de distribución, concebir el sistema  como un sistema abierto y ofrecer escalabilidad. | V |
| 2. | La transparencia de acceso oculta las diferencias en la representación de los datos y en cómo se accede a los recursos. | V |
| 3. | Cuando un sistema distribuido es abierto facilita que uno de sus módulos o componentes pueda utilizarse en otro sistema distribuido. | V |
| 4. | La *capa de middleware*, que se ubica bajo el nivel de aplicación, puede integrar algunos mecanismos de comunicación que faciliten la  programación de aplicaciones distribuidas. | V |
| 5. | El uso de algoritmos descentralizados permite ofrecer transparencia de acceso y de ubicación, al estar distribuida la carga del algoritmo entre diferentes nodos. | F |
| 6. | La replicación y el “caching” no deben utilizarse para incrementar la escalabilidad de un sistema distribuido cuando la mayoría de los accesos  que se realizan son de consulta. | F |

# [2º PARCIAL 2016]



Sobre los sistemas distribuidos:

|  |  |  |
| --- | --- | --- |
| 7. | Los algoritmos descentralizados facilitan distribuir la carga computacional entre diferentes ordenadores. | V |
| 8. | Se requiere utilizar algoritmos descentralizados para conseguir escalabilidad administrativa, de modo que los cómputos se distribuyen  entre diferentes áreas administrativas del sistema. | F |
| 9. | Para mejorar la escalabilidad de tamaño, los clientes deben delegar en el  servidor tantas responsabilidades como sea posible. | F |
| 10. | El que ningún nodo mantenga toda la información completa es una característica de los algoritmos descentralizados. | V |
| 11. | La transparencia de reubicación requiere transparencia de migración. | V |
| 12. | Al proporcionar transparencia de replicación, también se proporcionará transparencia de ubicación. | V |

## [FINAL 2015]

Sobre los sistemas distribuidos:

|  |  |  |
| --- | --- | --- |
| 13 | La disponibilidad, una de las características de los sistemas distribuidos, sirve para ocultar las diferencias de los mecanismos de comunicación.  *JUSTIFICACIÓN: La disponibilidad implica que los servicios deben estar siempre disponibles (está directamente relacionada con el fallo del sistema, no con la transparencia de acceso).* | F |
| 14 | La transparencia de replicación oculta la coordinación entre las actividades que gestionan un conjunto de recursos para mantener su consistencia.  *JUSTIFICACIÓN: La definición dada se corresponde con la transparencia de transacción.* | F |
| 15 | Los sistemas distribuidos basados en Active Directory proporcionan escalabilidad administrativa. | V |
| 16 | La capa de middleware ayuda a conseguir los cuatro objetivos principales de los sistemas distribuidos: descentralización, replicación, transparencia de acceso y escalabilidad.  *JUSTIFICACIÓN: Los objetivos indicados son incorrectos. Los cuatro objetivos son: facilitar acceso*  *a los recursos remotos, transparencia de distribución, sistema abierto y escalabilidad.* | F |

Respecto a los algoritmos descentralizados:

|  |  |  |
| --- | --- | --- |
| 17. | Si hay un nodo que mantiene toda la información relevante de un algoritmo, pero este nodo toma decisiones basadas en su conocimiento local, entonces el algoritmo es descentralizado.  *JUSTIFICACIÓN: Para que sea descentralizado, ningún nodo debe mantener toda la información completa que necesite el algoritmo.* | F |
| 18. | La utilización de algoritmos descentralizados permite ofrecer transparencia de acceso y de ubicación, al estar distribuida la carga del algoritmo entre diferentes nodos.  *JUSTIFICACIÓN: Esta distribución de la carga del algoritmo por sí misma no ofrece los tipos de*  *transparencia indicados.* | F |
| 19. | La utilización de algoritmos descentralizados permite mejorar la escalabilidad de distancia. | V |
| 20. | Para mejorar la escalabilidad de tamaño se suele utilizar distribución de responsabilidades, replicación, caching y algoritmos descentralizados. | V |



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

# [2º PARCIAL 2015]

Sobre los sistemas distribuidos:

|  |  |  |
| --- | --- | --- |
| 21. | La transparencia de acceso oculta las diferencias en la representación de los datos y en cómo se accede a los recursos. | V |
| 22. | Cuando un sistema distribuido es abierto facilita que uno de sus módulos o componentes  pueda utilizarse en otro sistema distribuido. | V |
| 23. | Para conseguir escalabilidad de distancia debe asumirse que se está utilizando una red de área local.  JUSTIFICACIÓN: *La escalabilidad de distancia permite extender el sistema por redes de área amplia (WAN), por lo que si se usan algoritmos basados en redes de área local, se debe considerar que no se está usando una red local, sino una red WAN, por lo que hay que tener en cuenta los efectos de los*  *retardos en la transmisión de los datos y la menor fiabilidad de las comunicaciones.* | F |
| 24. | Los algoritmos descentralizados facilitan distribuir la carga computacional entre diferentes ordenadores. | V |
| 25. | Se requiere utilizar algoritmos descentralizados para conseguir escalabilidad administrativa, de modo que los cómputos se distribuyen entre diferentes áreas administrativas del sistema.  JUSTIFICACIÓN: *Para conseguir escalabilidad administrativa se deben utilizar protocolos y mecanismos estándar de autenticación y autorización; así como implementar mecanismos para proteger a cada organización del resto y del propio sistema.* | F |
| 26. | Para mejorar la escalabilidad de tamaño, los clientes deben delegar en el servidor tantas responsabilidades como sea posible.  JUSTIFICACIÓN: *Al contrario, contra menos se centralicen las tareas en los servidores, mayor escalabilidad de tamaño se podrá conseguir.* | F |
| 27. | La *capa de middleware*, que se ubica bajo el nivel de aplicación, puede integrar algunos  mecanismos de comunicación que faciliten la programación de aplicaciones distribuidas; por ejemplo: JMS. | V |

# [FINAL 2014]

Sobre los sistemas distribuidos:

|  |  |
| --- | --- |
| 31. La transparencia de ubicación oculta el hecho de que un recurso esté ubicado en memoria volátil o en memoria persistente. | F |
| 32. Para mejorar la escalabilidad administrativa hay que conseguir que todos los  ordenadores del sistema pertenezcan a una única organización. | F |
| 33. La capa de middleware puede integrar algunos mecanismos de comunicación de alto nivel que faciliten la programación de aplicaciones distribuidas; por  ejemplo: RPC. | V |
| 34. Las arquitecturas de sistema para sistemas distribuidos representan la implementación física de los componentes software del sistema en máquinas  reales. | V |

# [2º PARCIAL 2014]

1. Sobre los sistemas distribuidos:

|  |  |
| --- | --- |
| F | A.- Todos los sistemas de tiempo real son ejemplos de sistemas distribuidos. |
| V | B.- Algunos tipos de transparencia pueden comprometer la eficiencia del sistema, por  ejemplo introduciendo retardos en las interacciones entre componentes de un sistema distribuido. |
| V | C.- Un sistema distribuido ofrece la imagen de un sistema coherente y único. |
| F | D.- Los sistemas distribuidos proporcionan diferentes tipos de transparencia. Entre |



|  |  |
| --- | --- |
|  | ellos: transparencia en el rendimiento, en la escalabilidad, en su disponibilidad, en la seguridad... |
| F | E.- Si se desea que un sistema distribuido ofrezca escalabilidad de tamaño, no se  debe hacer uso de técnicas de replicación, ya que al actualizar el estado de un componente tendremos que propagar tal actualización a todas las réplicas. |