Ejercicios de examen de arquitectura de los sistemas distribuidos

Febrero del 2017. Grupo de mañana.

La empresa *miCaja* ofrece un servicio bancario que permite a sus clientes gestionar sus cuentas por Internet, exceptuando ciertas operaciones (como la creación de una nueva cuenta) que deben hacerse de forma presencial. Como en cualquier banco, un cliente puede ser titular de varias cuentas y una misma cuenta puede tener múltiples titulares. Dentro de la organización, existen empleados que desempeñan el rol de gestores, realizando ofertas ventajosas a los titulares de cuentas (por ejemplo, un regalo, por su fidelidad, a los titulares de una cuenta que lleva abierta 50 años). Cada gestor dispone de una aplicación G que le permite (OP1) realizar una oferta introduciendo el código de la cuenta a la que va destinada y el mensaje asociado a la misma. Los clientes usan una aplicación C que les ofrece, entre otras, las siguientes operaciones: (OP2) *login*, que permite al cliente iniciar una sesión; (OP3) *consulta*, que, durante una sesión, deja al cliente conocer el saldo de una de sus cuentas; (OP4) *transferencia*, que permite que un cliente, durante una sesión, pueda realizar una transferencia de dinero desde una de sus cuentas hasta otra cuenta (que puede ser o no suya y que, incluso, puede corresponder a otra entidad bancaria); (OP5) *logout*, que termina una sesión. Mientras esté en una sesión, el cliente será notificado de todas las ofertas que se vayan emitiendo en las que esté involucrado, así como de los cambios en el saldo de cualquiera de sus cuentas. Se usa una arquitectura editor/subscriptor para la gestión de las notificaciones y una solución cliente/servidor para el resto de la funcionalidad. Por motivos de seguridad, se almacenan 3 copias independientes de los datos de cada cuenta bancaria. Cualquier operación que modifique la información asociada a una cuenta (por ejemplo, el saldo) no se completará hasta que estén actualizadas las tres copias. Por otro lado, si momentáneamente se deja de tener acceso a una de las copias de una cuenta, por seguridad, no se permite ninguna operación sobre la misma. Por último, dado que puede haber problemas de concurrencia al actualizar el saldo de una cuenta debido a la existencia de múltiples titulares, se usa un servicio remoto de cerrojos (SC) para solventarlos, estando replicado dicho servicio para aumentar la fiabilidad del sistema.

1. ¿Qué módulos realizan el papel de subscriptores?
   a. **Sólo C**
   b. Sólo G
   c. C y G
   d. C y SC

Explicación

El módulo C realiza el rol de subscriptor puesto que está interesado en ser notificado de las ofertas que realizan los gestores a cualquiera de las cuentas de las que es titular el cliente, así como de los cambios en el saldo de dichas cuentas.

2. ¿Qué módulos realizan el papel de editores?
   a. **C y G**
   b. Sólo G
   c. Sólo C
   d. C y SC

Explicación

El módulo G desempeña el papel de editor puesto que cada vez que un gestor realiza una oferta a una cuenta deben ser informados todos los titulares de la misma. Por su parte, el módulo C también tiene asociado ese rol ya que cada vez que un cliente realiza una transferencia en la que está involucrada una de sus cuentas, todos los titulares de esa cuenta son notificados del cambio de saldo correspondiente.

3. ¿Cuántos temas existen en el esquema editor/subscriptor?
   a. **Uno por cada cuenta**
   b. Uno por cada cliente
   c. Uno por cada gestor
   d. Uno por cada operación

Explicación

Tanto los mensajes de oferta como las operaciones que conllevan un cambio de saldo están vinculadas a una determinada cuenta y, por tanto, ese elemento es el que representa a un tema en el esquema editor/subscriptor planteado.

4. Suponga que en un momento dado hay sesiones de X clientes tal que cada uno de ellos es titular de Y cuentas. ¿Cuántas subscripciones existen en el sistema?
   a. **X * Y**
   b. Y
   c. X
   d. X + Y

Explicación

Durante una sesión el cliente tiene que estar subscrito a todas sus cuentas para poder recibir las notificaciones, tanto de ofertas como de cambios de saldo, asociadas a las mismas. Por tanto, habrá en total X * Y subscripciones.

5. ¿Qué acción editor/subscriptor implica OP1?
   a. **publicación**
   b. baja
   c. subscripción
   d. ninguna

Explicación

Un mensaje de oferta de un gestor dirigido a una determinada cuenta se corresponde con una publicación en un esquema editor/subscriptor.

6. ¿Qué acción editor/subscriptor implica OP2?
   a. **subscripción**
   b. baja
   c. publicación
   d. ninguna

Explicación

Cuando un cliente inicia una sesión, el módulo C debe subscribirse a todas las cuentas de las que es titular dicho cliente.

7. ¿Qué acción editor/subscriptor implica OP3?
   a. **ninguna**
   b. baja
   c. publicación
   d. subscripción

Explicación

La consulta es una operación cliente/servidor y, puesto que no modifica el saldo de la cuenta, no implica ninguna acción en el esquema editor/subscriptor.

8. ¿Qué acción editor/subscriptor implica OP4?
   a. **publicación**
   b. baja
   c. subscripción
   d. ninguna

Explicación

La operación de transferencia entre cuentas implica la modificación de los saldos de las cuentas involucradas, provocando, por tanto, una acción de publicación por cada una de ellas (en el caso de la

cuenta destino, sólo se producirá una publicación si pertenece a esta entidad bancaria) que será notificada a sus titulares.

9. ¿Qué acción editor/subscriptor implica OP5?
    a. **baja**
    b. subscripción
    c. publicación
    d. ninguna

Explicación

Cuando un cliente finaliza una sesión, se eliminarán las subscripciones a sus cuentas.

10. ¿De qué tipo de sistema se trata según el teorema CAP?
    a. **CP**
    b. AP
    c. CA
    d. CAP

Explicación

Dado que cuando se pierde la accesibilidad a todas las copias de los datos de una cuenta por problemas de conectividad (P) no se llevan a cabo operaciones sobre esa cuenta, se está prefiriendo la consistencia (C) sobre la disponibilidad (A).

11. ¿De qué tipo de sistema se trata según el teorema PACELC?
    a. **PCEC**
    b. PAEC
    c. PAEL
    d. PCEL

Explicación

Por un lado, cuando hay problemas de conectividad, se prefiere tener consistencia frente a disponibilidad (PC). Por otro lado, cuando el sistema tiene un funcionamiento normal, se opta por la consistencia frente a la latencia (EC) ya que la operación de escritura no se completa hasta que todas las copias estén actualizadas, lo que resulta en una operación más lenta pero que evita que puedan leerse datos obsoletos.

12. ¿Son idempotentes OP3 y OP4?
    a. **Sólo OP3**
    b. Sólo OP4
    c. Ambas
    d. Ninguna

Explicación

La operación de consulta de saldo (OP3) es idempotente: el resultado de una operación es el mismo aunque ésta se repita de forma inadvertida. Sin embargo, la operación de transferencia no lo es, puesto que si se volviera a ejecutar inadvertidamente, se aplicaría nuevamente el reajuste de saldos en las cuentas involucradas.

13. ¿Qué módulo debería enviar los mensajes de renovación del *lease*, y a qué otro módulo, si se aplica un esquema de *leasing* al servicio de cerrojos?
    a. **De C a SC**
    b. De SC a C
    c. De G a SC
    d. De SC a G

Explicación

Para evitar que un cerrojo pueda quedar indefinidamente cerrado en caso de que el proceso que lo posee se caiga, se puede aplicar la técnica del *leasing*, haciendo que el proceso cliente (C) que lo posee solicite periódicamente al servidor de cerrojos (SC) la renovación del *lease*.

14. ¿Qué esquema de replicación de servicio sería más adecuado si se pretende reducir el tiempo de recuperación del servicio de cerrojos?
    a. **Activa**
    b. *hot standby*
    c. *warm standby*
    d. *cold standby*

Explicación

Con el esquema activo de replicación de servicios, todos los servidores que toman parte del servicio replicado procesan las peticiones de los clientes, por lo que si uno de ellos se cae, la recuperación es instantánea.

15. Se plantea usar un esquema con un filtro de eventos por contenido en vez de un filtro por temas. ¿Para cuál de estos casos ese cambio sería más ventajoso en el sentido de reducir el número de mensajes recibidos pero no deseados?
    a. **Interés en conocer todas las ofertas que realiza un determinado gestor.**
    b. Interés en conocer todas las ofertas que se producen en el sistema.
    c. Interés en conocer qué cambios de saldo se producen en un determinado conjunto de cuentas.
    d. Interés en conocer qué ofertas recibe un determinado conjunto de cuentas.

Explicación

Veamos cada caso planteado analizando en cuál el uso de un filtro por contenido sería más efectivo (es decir, descartaría más eventos no deseados).

- En el caso de estar interesado en conocer todas las ofertas que se producen en el sistema, el subscriptor tendría que suscribirse a todas las cuentas y descartar los eventos que estén vinculados a cambios de saldo.
- En cuanto al caso de estar interesado en conocer qué cambios de saldo se producen en un determinado conjunto de cuentas, el subscriptor tendría que suscribirse a esas cuentas y descartar los eventos que estén vinculados a ofertas.
- Por lo que se refiere al caso de estar interesado en conocer qué ofertas reciben un determinado conjunto de cuentas, el subscriptor tendría que suscribirse a esas cuentas y descartar los eventos que estén vinculados a cambios de saldo.
- Con respecto al caso de estar interesado en conocer todas las ofertas que realiza un determinado gestor, el subscriptor tendría que suscribirse a todas las cuentas y descartar tanto los eventos que estén vinculados a cambios de saldo como los correspondientes a ofertas realizadas por otros gestores. Se trata, por tanto, del caso donde sería más ventajoso usar un filtro por contenido.

Febrero del 2017. Grupo de tarde.

Se pretende implementar un servicio de log global para un sistema distribuido basado en un esquema similar al usado en UNIX, donde cada mensaje de log tiene asociada una etiqueta con el siguiente formato: `facility.priority`. La `facility` corresponde al componente o subsistema al que está asociado el mensaje (por ejemplo, el correo, la web, el sistema de impresión o el `kernel`, hasta un total de 32 distintas) y la `priority` al grado de urgencia (8 niveles que van, de menor a mayor urgencia, desde `info` hasta `panic`). Tomando como ejemplo la `facility` web, todas las aplicaciones relacionadas con este servicio (como, por ejemplo, el navegador NW o el servidor web SW) escribirán mensajes de log especificando esta `facility` y el nivel de urgencia correspondiente. Por otro lado, el administrador del sistema debe especificar qué tratamiento se realiza con cada tipo de mensaje de log (por ejemplo, escribirlos en la consola del administrador, enviárselos por correo o imprimirlos). Para ello, el administrador desplegará una serie de aplicaciones que recibirán como información de configuración el tipo de mensajes de log que gestionan y que realizarán un determinado tratamiento con ellos (por ejemplo, CO los escribe en consola, MX los envía por correo y PR los imprime). Nótese que un mismo mensaje de log puede ser tratado por varias aplicaciones. Asimismo, téngase en cuenta que una aplicación de tratamiento de un cierto tipo de mensajes de log también puede necesitar generar a su vez mensajes de log (por, ejemplo, si la impresora no responde, la aplicación PR puede generar un mensaje de log asociado a la `facility` del sistema de impresión).

Evidentemente, para evitar situaciones patológicas, el administrador no debería configurar que los mensajes de log asociados a una determinada `facility` sean tratados por una aplicación que usa esa `facility`. Para implementar este sistema, se utilizará una arquitectura editor/subscriptor con temas jerárquicos (2 niveles) y uso de comodines. Por motivos de trazabilidad y auditoría, todo mensaje de log será también guardado en un sistema de almacenamiento SA. La operación de escritura en SA seguirá un modelo cliente/servidor y siempre se añadirá al final del fichero (no escribe en la posición correspondiente al *offset* actual del proceso escritor sino que lo añade siempre al final del fichero). Por razones de seguridad, se almacenan varias copias de los datos en SA, existiendo una copia primaria. La operación de escritura se completará después de modificar la copia primaria, propagándose posteriormente a las otras copias. Las operaciones de lectura, que realizan procesos auditores, pueden acceder a cualquier copia. Por otro lado, si momentáneamente se deja de tener acceso a una de las copias, se seguirán actualizando las copias accesibles (una de las cuales será considerada como primaria) realizando un proceso de consolidación cuando se recupere la accesibilidad. Por último, dado que puede haber procesos de auditoría accediendo también a SA, se usará un servicio remoto de cerrojos (SC) para solventar los problemas de concurrencia, estando replicado dicho servicio para aumentar la fiabilidad.

1. ¿Qué rol realiza el módulo NW?
    a. **Sólo Ed**
    b. Sólo Su
    c. Ed y Su
    d. Ninguno

Explicación

El módulo NW realiza el rol de editor ya que genera mensajes de log que serán notificados a todos los procesos que están configurados para tratar ese tipo de mensajes de log.

2. ¿Qué rol realiza el módulo PR?
    a. **Ed y Su**
    b. Sólo Su
    c. Sólo Ed
    d. Ninguno

Explicación

El módulo PR se encarga de tratar los tipos de mensajes de log para los que está configurado, enviándolos a la impresora, por lo que actúa de subscriptor al tener que ser notificado de los mismos. Asimismo, este módulo también realiza el rol de editor ya que genera mensajes de log cuando se encuentra alguna situación anómala al interactuar con la impresora.

3. ¿Cuántos temas existen en el esquema editor/subscriptor?
    a. **32 * 8**
    b. 32
    c. 8
    d. 32 + 8

Explicación

Para cada `facility` hay 8 niveles de prioridad. Dado que el administrador puede especificar qué tratamiento recibe un mensaje de log que corresponda a una determinada `facility` y a un cierto nivel de prioridad, habrá un tema por cada combinación posible, resultando en un total de 32 multiplicado por 8.

4. ¿Cuántas operaciones de subscripción se requieren para procesar todos los mensajes `info` y `panic` si se usan comodines?
    a. **2**
    b. 16
    c. 64
    d. 40

Explicación

Gracias al uso de comodines se puede especificar en una sola operación de subscripción el interés por todos los mensajes de un mismo nivel de prioridad (por ejemplo, `*/info`) o de una determinada `facility` con independencia de cuál sea su nivel de prioridad (por ejemplo, `web/*`). Para el enunciado planteado, harán falta dos subscripciones gracias al uso de comodines: `*/info` y `*/panic`.

5. ¿Qué acción genera NW cuando detecta un error interno?
    a. **publicación**
    b. baja
    c. subscripción
    d. notificación

Explicación

Escribe un mensaje de log, lo que corresponde a una publicación que debe ser notificada a todos los procesos subscritos a ese tipo de mensaje.

6. ¿Qué acción realiza CO en su fase inicial?
    a. **subscripción**
    b. baja
    c. publicación
    d. notificación

Explicación

Se tiene que subscribir a todos los tipos de mensajes de log especificados en su configuración.

7. ¿Qué acción realiza CO en su fase final?
    a. **baja**
    b. subscripción
    c. publicación
    d. notificación

Explicación

Se tiene que dar de baja de todos los tipos de mensajes de log especificados en su configuración.

8. ¿Qué acción realiza PR cuando detecta un error en la impresora?
    a. **publicación**
    b. baja
    c. subscripción
    d. notificación

Explicación

Escribe un mensaje de log, lo que corresponde a una publicación que debe ser notificada a todos los procesos subscritos a ese tipo de mensaje.

9. ¿Qué acción recibida por MX causa que envíe un mensaje?
    a. **notificación**
    b. subscripción
    c. publicación
    d. baja

Explicación

Ese proceso envía un correo al administrador cuando es notificado de un mensaje de log que corresponde a uno de los tipos de mensajes especificados en su configuración.

10. ¿De qué tipo de sistema se trata según el teorema CAP?
    a. **AP**
    b. CP
    c. CA
    d. CAP

Explicación

Dado que cuando se pierde la accesibilidad a todas las copias de los datos de una cuenta por problemas de conectividad (P) se siguen permitiendo las actualizaciones de las copias accesibles, se está prefiriendo la disponibilidad (A) sobre la consistencia (C).

11. ¿De qué tipo de sistema se trata según el teorema PACELC?
    a. **PAEL**
    b. PAEC
    c. PCEC
    d. PCEL

Explicación

Por un lado, cuando hay problemas de conectividad, se prefiere tener disponibilidad frente a consistencia (PA). Por otro lado, cuando el sistema tiene un funcionamiento normal, se opta por la latencia frente a la consistencia (EL) ya que la operación de escritura se completa en cuanto esté actualizada la copia primaria, lo que resulta en una operación más eficiente pero que puede conducir a que se lean datos obsoletos.

12. ¿Es idempotente y *stateless* la escritura sobre SA?
    a. **No; Sí.**
    b. Sí; No.
    c. No; No.
    d. Sí; Sí.

Explicación

La operación de escritura al final del fichero no es idempotente puesto que si se repite inadvertidamente se almacenará repetidamente el mismo mensaje de log. Sin embargo, sí es *stateless* puesto que no requiere para su procesamiento de un estado volátil en el servidor. De esta manera, aunque se reinicie un servidor, puede procesar sin problemas una petición de este tipo pues depende solamente del tamaño del fichero que es un valor persistente.

13. ¿Qué módulo debería enviar los mensajes de renovación del *lease* si se aplica un esquema de *leasing* al editor/subscriptor?
    a. **CO**
    b. SW
    c. SC
    d. SA

Explicación

Si se aplica un esquema de *leasing* a una solución editor/subscriptor, los procesos que siguen el rol de subscriptores deben renovar el *lease*, de manera que si en un momento dado un subscriptor se cae, se puede dar de baja de todos los temas que estuviera subscrito. De los procesos planteados en la pregunta, sólo CO actúa como subscriptor.

14. ¿Qué esquema de replicación de servicio sería más adecuado si se quiere minimizar el gasto de recursos de procesamiento (como, por ejemplo, consumo de energía) mientras el sistema no tenga errores aunque se sacrifique el tiempo de recuperación del servicio de cerrojos?
    a. ***cold standby***
    b. *hot standby*
    c. *warm standby*
    d. Activa

Explicación

En el esquema *cold standby*, hay un servidor primario que procesa las peticiones de los clientes y almacena su estado de forma persistente, mientras que el resto de los servidores están desactivados y sólo entran en acción cuando se detecta que el primario está caído. Esta solución minimiza el gasto de recursos de procesamiento mientras el sistema no tenga errores, ya que en las otras planteadas los servidores realizan procesamiento aunque el sistema esté completamente operativo. La contrapartida es que se obtiene un peor tiempo de recuperación ya que, ante la caida del primario, hay que activar un nuevo servidor primario y éste tiene que reconstruir su estado leyendo del almacenamiento persistente.

15. Se plantea usar un esquema con un filtro de eventos por contenido en vez de un filtro por temas. ¿Para cuál de estos casos ese cambio sería más ventajoso en el sentido de reducir el número de mensajes recibidos pero no deseados?
    a. **Interés en tratar los mensajes generados por una aplicación que usa todas las `facilities`.**
    b. Interés en tratar todos los mensajes del sistema.
    c. Interés en tratar todos los mensajes de tipo `panic`.
    d. Interés en tratar los mensajes de tipo `panic` o `info` que genera NW.

Explicación

Veamos cada caso planteado analizando en cuál el uso de un filtro por contenido sería más efectivo (es decir, descartaría más eventos no deseados).

- En el caso de estar interesado en tratar todos los mensajes de log, el subscriptor tendría que suscribirse a todos los tipos de mensajes (para lo cual podría usar una subscripción con comodines: `*/*`) y no descartar ninguno.
- En cuanto al caso de estar interesado en tratar todos los mensajes de tipo `panic`, el subscriptor tendría que subscribirse a los mensajes de ese tipo de todas las `facilities` (para lo cual podría usar una subscripción con comodines: `*/panic`) y no descartar ninguna notificación.
- Por lo que se refiere al caso de estar interesado en tratar los mensajes de tipo `panic` o `info` generados por NW, el subscriptor tendría que subscribirse a los mensajes de ambas prioridades de todas las `facilities` (para lo cual podría usar dos subscripciones con comodines: `*/panic` y `*/info`) y descartar todos los mensajes generados por procesos que no sean NW. También se podría suponer que NW sólo genera mensajes de la `facility` web y, en ese caso, sólo haría falta subscribirse a `web/panic` y `web/info` y descartar los mensajes generados por procesos que no sean NW.
- Con respecto al caso de estar interesado en tratar los mensajes generados por una aplicación que usa todas las `facilities`, el subscriptor tendría que suscribirse a todos los tipos de mensajes (para lo cual podría usar una subscripción con comodines: `*/*`) y descartar todos los que no haya generado esa aplicación. Se trata, por tanto, del caso donde sería más ventajoso usar un filtro por contenido.

Marzo del 2016. Grupo de tarde.

La empresa *GameOver* proporciona la infraestructura para el despliegue de juegos *online* multijugador de gran escala (*Massively Multiplayer Online Game*) basados en un universo de *habitaciones* conectadas por *puertas*. Esta infraestructura está implementada en la empresa (módulo G) mediante una solución que combina un esquema cliente/servidor, para la descarga de información, con un editor/subscriptor, para la actualización automática de la posición de los jugadores. En este sistema hay dos tipos de usuarios: los jugadores, que deambulan por el universo virtual, y los árbitros, que contemplan las acciones de los diversos jugadores, sin intervenir en el universo virtual. La empresa proporciona una aplicación (C) que permite a un usuario crear un nuevo juego definiendo cada una de las habitaciones del universo y su conectividad. Una vez completada la fase de definición, toda esa información, de gran volumen, se carga en la infraestructura de la empresa. Finalizada esa etapa, el proceso creador se convierte en el árbitro inicial del sistema, pudiéndose incorporar progresiva y dinámicamente más árbitros a lo largo de una partida. Un árbitro contemplará en su pantalla todo el universo y conocerá en todo momento qué jugadores están ubicados en todas y cada una de las habitaciones. La aplicación que corresponde a un árbitro (A) dispone de dos operaciones: (OP1) inicio de la supervisión, que, entre otras acciones, descarga del servidor G todas las imágenes del universo virtual y descarga desde cualquier otro árbitro la información sobre qué jugadores están ubicados en todas y cada una de las habitaciones (el árbitro inicial no necesita descargar esa información de ubicación) mostrando en la pantalla toda esta información, momento a partir del cual esta aplicación irá recibiendo automáticamente todas las actualizaciones de la ubicación de los jugadores, que reflejará en la pantalla; (OP2) fin de la supervisión. Un jugador ve en su pantalla sólo la habitación que ocupa en ese instante y, por tanto, sólo debe conocer en cada momento qué otros jugadores están en esa

misma habitación. La aplicación que corresponde a un jugador (J) dispone, entre otras, de las siguientes operaciones: (OP3) *login* y (OP4) *logout*, que contactan con el servidor de la empresa para proporcionar información del jugador y para finalizar el juego, respectivamente; (OP5) entrar en una determinada habitación, que, entre otras acciones, descarga desde el servidor G la imagen de la habitación y desde cualquier árbitro la información sobre qué jugadores están ubicados en esa habitación en ese momento mostrando en la pantalla esa información (esta entrada se reflejará automáticamente en las pantallas de todos los árbitros y de los otros jugadores que están en esa nueva habitación), momento a partir del cual esta aplicación irá recibiendo automáticamente todas las actualizaciones que corresponden a otros jugadores que entran y salen de la habitación, que reflejará en su pantalla; (OP6) salir de la habitación que ocupa el jugador, que elimina de la pantalla la representación de la habitación, y que se reflejará automáticamente en las pantallas de todos los árbitros y de los otros jugadores que estaban en esa habitación. Nótese que la acción de atravesar la puerta entre dos habitaciones se interpreta como una operación de salir seguida por una de entrar.

1. ¿Qué módulos realizan el papel de subscriptores?
    a. **J y A**
    b. J
    c. A
    d. G

Explicación

El módulo J correspondiente a un determinado jugador realiza el rol de subscriptor puesto que está interesado en ser notificado cada vez que otro jugador entra o sale de la habitación que este primero ocupa actualmente. El módulo A también hace ese mismo papel de subscriptor ya que necesita conocer en todo momemto las ubicaciones de todos los jugadores.

2. ¿Qué módulos realizan el papel de editores?
    a. **J**
    b. J y A
    c. A
    d. G

Explicación

El módulo J realiza el rol de editor puesto que los cambios de ubicación del jugador asociado a ese módulo deben ser notificados a todos los árbitros así como a todos los jugadores que estén en la misma habitación que este jugador.

3. ¿Qué acción implica OP1?
    a. **subscripción**
    b. baja
    c. publicación
    d. notificación

Explicación

El inicio de la supervisión por parte de un árbitro conlleva la subscripción a todos los temas asociados con todas y cada una de las habitaciones puesto que necesita conocer en todo momento el estado del sistema.

4. ¿Qué acción implica OP2?
    a. **baja**
    b. subscripción
    c. publicación
    d. notificación

Explicación

El fin de la supervisión por parte de un árbitro conlleva la baja de la subscripción a todos los temas del sistema puesto que ya necesita conocer ninguna información acerca del mismo.

5. ¿Las acciones OP5 y OP6 provocan una subscripción?
    a. **OP5 sí**
    b. OP6 sí
    c. Ambas
    d. Ninguna

Explicación

Cuando un jugador entra en una habitación, debe subscribirse al tema asociado a la misma para ser notificado a partir de ese momento de que otros jugadores entran y salen de esa habitación.

6. ¿Las acciones OP5 y OP6 causan baja de una subscripción?
    a. **OP6 sí**
    b. OP5 sí
    c. Ambas
    d. Ninguna

Explicación

Cuando un jugador sale de una habitación, debe darse de baja del tema asociado a la misma para dejar de ser notificado cuando otros jugadores entran y salen de la misma.

7. ¿Las acciones OP5 y OP6 provocan publicar un evento?
    a. **Ambas**
    b. OP5 sí
    c. OP6 sí
    d. Ninguna

Explicación

Tanto la operación de entrar en una habitación como la de salir de la misma deberán ser comunicadas a todos los árbitros así como a los usuarios que comparten esa habitación para que puedan reflejar en sus respectivas pantallas ese hecho. Ambas acciones, por tanto, corresponden a una operación de publicación.

8. ¿Qué mensaje recibido por un jugador provoca que se elimine a otro jugador de su pantalla?
    a. **notificación**
    b. baja
    c. subscripción
    d. publicación

Explicación

El borrado en la pantalla de un jugador de la imagen de otro se lleva a cabo cuando se recibe la notificación de que ese segundo jugador ha salido de la habitación que compartían hasta ese momento.

9. ¿Qué mensaje recibido por un jugador provoca que se añada otro jugador en su pantalla?
    a. **notificación**
    b. baja
    c. subscripción
    d. publicación

Explicación

El dibujar en la pantalla de un jugador la imagen de otro se lleva a cabo cuando se recibe la notificación de que ese segundo jugador ha entrado en la habitación donde está ubicado el primero.

10. Teniendo en cuenta la funcionalidad de J, ¿Cuántos temas existen en el esquema editor/subscriptor?
    a. **Uno por habitación**
    b. Sólo uno
    c. Uno por árbitro
    d. Uno por jugador

Explicación

Un jugador sólo debe ser informado cada vez que otro jugador entra o sale de la habitación donde está ubicado el primero. Por tanto, hay tantos temas como habitaciones.

11. Se requiere saber cuánto tiempo participa en el juego cada jugador. Para ello, se calcula la diferencia del tiempo tomado en el servidor en OP3 y OP4. Se están valorando dos implementaciones: (1) en OP3 G envía el valor de tiempo a J y éste presenta ese valor original a G en OP4 para calcular la diferencia con el tiempo actual; (2) en OP3 G almacena el tiempo en ese momento y le devuelve a J un identificador de sesión que éste presenta a G en OP4 lo que le permite recuperar el valor original y restarlo del tiempo actual. ?`Qué solución (i) proporciona mayor tolerancia a los reinicios del servidor y cuál (ii) facilita el reparto de carga entre servidores si se usa un esquema de múltiples servidores en G?
    a. **(1)(1)**
    b. (2)(2)
    c. (1)(2)
    d. (2)(1)

Explicación

La solución (2) requiere estado en el servidor puesto que éste tiene que almacenar el momento en el que el usuario ha hecho el *login*. La solución (1) no requiere estado en el servidor ya que en la operación de *logout* éste recibe la marca de tiempo inicial. Por tanto, la solución (1), al no requerir estado en el servidor, proporciona mayor tolerancia a los reinicios del servidor y facilita el reparto de carga entre servidores si se usa un esquema de múltiples servidores.

12. La carga al servidor de las imágenes del universo desde C toma un tiempo considerable y durante la misma puede perderse la conectividad con la empresa, sobretodo si el usuario creador utiliza un dispositivo móvil. Ante este problema, cuando se recupera la conectividad, se usa una petición PUT de HTTP con rangos que permite enviar un determinado intervalo de bytes de un recurso web. ¿Se trata de una operación (i) idempotente, (ii) que requiere estado en el servidor?
    a. **sí; no**
    b. no; sí
    c. no; no
    d. sí; sí

Explicación

La operación PUT con rangos es idempotente puesto que, aunque se repita varias veces, el resultado final es el mismo. Además, esta operación no requiere estado en el servidor al contener toda la información necesaria para su procesamiento (la identificación del recurso y el rango de bytes escritos).

13. Suponiendo que se usa un esquema de *leasing* para mejorar la tolerancia a fallos del esquema editor/subscriptor, ¿qué módulos deben renovar el *lease*?
    a. **J y A**
    b. J
    c. A
    d. G

Explicación

Cuando se aplica un mecanismo de *leasing* en un esquema editor-subscriptor, son los subscriptores los encargados de enviar el mensaje de renovación. Por tanto, se trata de los módulos J y A.

14. ¿Qué módulo se beneficiaría más de una operación de subscripción múltiple con comodines?
    a. **A**
    b. J
    c. G
    d. C

Explicación

Un jugador sólo está suscrito a un tema/habitación en cada momento. Un árbitro, sin embargo, necesita estar suscrito a todos los temas/habitaciones por lo que se beneficiaría del uso de una operación de subscripción múltiple.

15. Si se usa un esquema de *binding*, ¿qué módulos deben darse de alta en el mismo?
    a. **G y A**
    b. J y A
    c. G y J
    d. J

Explicación

Todo módulo que sea el destinatario de una petición debe darse de alta en el servicio de binding. El módulo G lo requiere tanto en su rol de proceso intermediario en el esquema editor/subscriptor como en su papel de servidor a la hora de descargar las imágenes del universo del juego. El módulo A también lo requiere puesto que actúa como servidor al que se le solicita el estado de ocupación de una o más habitaciones.

16. ¿Para el seguimiento de cuál de las siguientes informaciones de estado sería más ventajoso usar un esquema con un filtro de eventos por contenido en vez de un filtro por temas en el sentido de reducir el número de notificaciones no deseadas?
    a. **Ubicación en cada momento de un jugador dado.**
    b. Qué jugadores hay en cada instante en un determinado conjunto de habitaciones.
    c. Conocer todo el estado del sistema.
    d. Interés en cuando cualquiera de un conjunto de jugadores entra en una determinada habitación.

Explicación

Veamos cada caso planteado analizando en cuál el uso de un filtro por contenido sería más efectivo (es decir, descartaría más eventos no deseados).

- En cuanto al caso de estar interesado en saber qué jugadores hay en cada instante en un determinado conjunto de habitaciones, el subscriptor tendría que suscribirse a los temas asociados a esas habitaciones y no descartar ningún evento.
- En el caso de estar interesado en conocer todo el estado del sistema, el subscriptor tendría que suscribirse a todas las habitaciones y no descartaría ningún evento.
- Con respecto al caso de estar interesado en ser notificado cuando cualquiera de un conjunto de jugadores entra en una determinada habitación, el subscriptor tendría que suscribirse al tema correspondiente a esa habitación y descartar todos los eventos que no estén vinculados con la entrada de uno de esos jugadores.
- Por lo que se refiere al caso de estar interesado en conocer la ubicación en cada momento de un jugador dado, el subscriptor tendría que suscribirse a todas las habitaciones y descartar todos los eventos que no correspondan a ese jugador. Nótese que en este caso, con un filtro por temas, el subscriptor tendría que recibir todos los eventos del sistema.

Marzo del 2016. Grupo de mañana.

La empresa *SoftStore* proporciona un servicio de distribución de aplicaciones destinado a dos tipos de personas: usuarios de las aplicaciones y desarrolladores de las mismas, organizados en equipos de desarrollo. Los usuarios instalan un programa U que proporciona las siguientes operaciones: (OP1) Búsqueda de aplicaciones basada en los criterios especificados por el usuario (nombre de la aplicación, equipo de desarrollo, temática, precio,...) mostrándose los resultados de la consulta paginados con botones para acceder a la página siguiente y anterior de los mismos; (OP2) Descarga e instalación de la aplicación seleccionada, tal que una vez instalada se muestra una ventana *popup* cada vez que hay una nueva versión de dicha aplicación por si el usuario quiere actualizarla; (OP3) Desinstalación de una aplicación; (OP4) Activación de un servicio de alertas que permite al usuario que lo activa conocer cuándo el equipo de desarrollo seleccionado publica una nueva aplicación (nueva, es decir, la primera versión de la aplicación) mostrándose una ventana *popup* para indicarlo y permitir descargarla e instalarla si lo considera oportuno. Los desarrolladores instalan un programa D que proporciona las siguientes operaciones: (OP5) Crear una nueva aplicación; (OP6) Actualizar una aplicación (es decir, crear una nueva versión de la misma). En la empresa está instalado el módulo software S. El servicio está implementado mediante un esquema cliente/servidor para las búsquedas, cargas y descargas, y dos esquemas editor/subscriptor independientes para el control de actualizaciones y las alertas de nuevas aplicaciones.

1. ¿Qué módulos realizan el papel de subscriptores en el control de actualizaciones?
    a. **U**
    b. D
    c. U y D
    d. S

Explicación
El módulo U realiza el rol de subscriptor del servicio de control de actualizaciones puesto que está interesado en ser notificado cada vez que la aplicación instalada es actualizada.

2. ¿Qué módulos realizan el papel de editores en el control de actualizaciones?
    a. **D**
    b. U
    c. U y D
    d. S

Explicación
El módulo D desempeña el papel de editor del servicio de control de actualizaciones ya que cada vez que un desarrollador crea una nueva versión de una aplicación deben ser informados todos los usuarios que la tienen instalada.

3. ¿Qué módulos realizan el papel de subscriptores en el servicio de alertas?
    a. **U**
    b. D
    c. U y D
    d. S

Explicación
El módulo U realiza el rol de subscriptor del servicio de alertas puesto que está interesado en ser notificado cada vez que un determinado equipo de desarrollo crea una nueva aplicación.

4. ¿Qué módulos realizan el papel de editores en el servicio de alertas?
    a. **D**
    b. U
    c. U y D
    d. S

Explicación
El módulo D desempeña el papel de editor del servicio de alertas ya que cada vez que un desarrollador de un determinado equipo crea una nueva aplicación deben ser informados todos los usuarios que están interesados en las aplicaciones de ese equipo de desarrollo.

5. ¿Qué acción implica OP2?
    a. **subscripción**
    b. baja
    c. publicación
    d. notificación

Explicación
La instalación de una aplicación conlleva la subscripción al tema asociado a dicha aplicación para poder recibir avisos cuando se actualiza.

6. ¿Qué acción implica OP3?
    a. **baja**
    b. subscripción
    c. publicación
    d. notificación

Explicación
La desinstalación de una aplicación implica la baja de la subscripción al tema asociado a dicha aplicación puesto que ya no se está interesado en las actualizaciones de la misma.

7. ¿Qué acción implica OP4?
    a. **subscripción**
    b. baja
    c. publicación
    d. notificación

Explicación
La activación del servicio de alertas conlleva la subscripción al tema asociado al equipo desarrollador correspondiente para poder recibir alertas cuando éste crea una nueva aplicación.

8. ¿Qué acción implica OP5?
    a. **publicación**
    b. baja
    c. subscripción
    d. notificación

Explicación La operación de crear una nueva aplicación por parte de un desarrollador de un determinando equipo deberá ser comunicada a todos los usuarios que han activado una alerta asociada a dicho equipo. Se trata, por tanto, de una operación de publicación.

9. ¿Qué acción implica OP6?
    a. **publicación**
    b. baja
    c. subscripción
    d. notificación

Explicación
La operación de crear una nueva versión de una aplicación deberá ser comunicada a todos los usuarios que la tienen instalada. Se trata, por tanto, de una operación de publicación.

10. ¿Qué acción implica el *popup* de actualizaciones?
    a. **notificación**
    b. baja
    c. subscripción
    d. publicación

Explicación
El *popup* de actualizaciones se muestra cuando llega al módulo U la notificación de la actualización.

11. ¿Qué acción implica el *popup* de alertas?
    a. **notificación**
    b. baja
    c. subscripción
    d. publicación

Explicación
El *popup* de alerta se muestra cuando llega al módulo U la notificación de la nueva aplicación.

12. ¿Cuántos temas existen en el esquema editor/subscriptor que gestiona las actualizaciones?
    a. **Uno por cada aplicación**
    b. Uno por cada usuario
    c. Uno por cada equipo de desarrollo
    d. Uno por cada actualización

Explicación

El servicio de control de actualizaciones permite ser informado cada vez que una aplicación se actualiza. Por tanto, hay un tema por cada aplicación existente.

13. ¿Cuántos temas existen en el esquema editor/subscriptor que gestiona las alertas?
    a. **Uno por cada equipo de desarrollo**
    b. Uno por cada usuario
    c. Uno por cada aplicación
    d. Uno por cada actualización

Explicación

El servicio de alertas permite ser informado cada vez que un equipo de desarrollo crea una nueva aplicación. Por tanto, hay un tema por cada equipo de desarrollo existente.

14. Se están valorando dos implementaciones del mecanismo de paginación de las respuestas: cuando se pulsa el botón siguiente o anterior, se envía en el mensaje (1) un parámetro que indica si se pide la siguiente o la anterior, respectivamente, o bien (2) el número absoluto de la página requerida. ¿Qué solución (i) proporciona mayor tolerancia a los reinicios del servidor y cuál (ii) facilita el reparto de carga entre servidores si se usa un esquema de múltiples servidores en S?
    a. **(2)(2)**
    b. (1)(1)
    c. (1)(2)
    d. (2)(1)

Explicación

La solución (1) requiere estado en el servidor puesto que éste tiene que almacenar el último número de página de resultados que ha presentado cada usuario para poder intepretar adecuadamente la petición de la página siguiente o previa. La solución (2) no requiere estado en el servidor al especificarse en la petición el número de página requerido. Por tanto, la solución (2), al no requerir estado en el servidor, proporciona mayor tolerancia a los reinicios del servidor y facilita el reparto de carga entre servidores si se usa un esquema de múltiples servidores.

15. La descarga de una aplicación toma un tiempo considerable y durante la misma puede perderse la conectividad con la empresa, sobretodo si el usuario utiliza un dispositivo móvil. Ante este problema, cuando se recupera la conectividad, se usa una petición GET de HTTP con rangos que permite solicitar un determinado intervalo de bytes de un recurso web. ¿Se trata de una operación (i) idempotente, (ii) que requiere estado en el servidor?
    a. **sí; no**
    b. no; sí
    c. no; no
    d. sí; sí

Explicación

La operación GET con rangos es idempotente puesto que, aunque se repita varias veces, el resultado final es el mismo. Además, esta operación no requiere estado en el servidor al contener toda la información necesaria para su procesamiento (la identificación del recurso y el rango de bytes solicitado).

16. Suponiendo que se usa un esquema de *leasing* para mejorar la tolerancia a fallos del esquema editor/subscriptor de alertas, ¿qué módulos deben renovar el *lease*?
    a. **U**
    b. D
    c. U y D
    d. S

Explicación

Cuando se aplica un mecanismo de *leasing* en un esquema editor-subscriptor, son los subscriptores los encargados de enviar el mensaje de renovación. Por tanto, se trata del módulo U.

17. Se plantea usar un esquema con un filtro de eventos por contenido en vez de un filtro por temas para el sistema de alertas. ¿Para cuál de estos casos ese cambio sería más ventajoso en el sentido de reducir el número de mensajes recibidos pero no deseados?
    a. **Interés en cuando se publica una aplicación de una determinada temática.**
    b. Interés en cualquier nueva aplicación.
    c. Interés en cuando un determinado equipo de desarrollo crea aplicaciones de un conjunto de temáticas.
    d. Interés en cuando cualquiera de un conjunto de equipos de desarrollo crea aplicaciones de una determinada temática.

Explicación

Veamos cada caso planteado analizando en cuál el uso de un filtro por contenido sería más efectivo (es decir, descartaría más eventos no deseados).

- En el caso de estar interesado en cualquier nueva aplicación, el subscriptor tendría que suscribirse a todos los equipos de desarrollo y no descartaría ningún evento.
- En cuanto al caso de estar interesado en ser notificado cuando un cierto equipo de desarrollo crea aplicaciones de un conjunto de temáticas, el subscriptor tendría que suscribirse al tema asociado a ese equipo y descartar todos los eventos que no correspondan a aplicaciones de esas temáticas.
- Con respecto al caso de estar interesado en ser notificado cuando cualquiera de un conjunto de equipos de desarrollo crea aplicaciones de una determinada temática, el subscriptor tendría que suscribirse a los temas correspondientes a esos equipos y descartar todos los eventos que no correspondan a aplicaciones de esa temática.
- Por lo que se refiere al caso de estar interesado en ser notificado cuando se publica una aplicación de una determinada temática, el subscriptor tendría que suscribirse a todos equipos y descartar todos los eventos que no correspondan a esa temática. Nótese que en este caso, con un filtro por temas, el subscriptor tendría que recibir todos los eventos del sistema.