

Esta prueba tiene un valor de 1 punto, y consta de 6 cuestiones tipo test. Cada cuestión plantea 4 alternativas y tiene una única respuesta correcta. Cada respuesta correcta aporta 1/6 puntos, y cada error descuenta 1/18 puntos. Debe contestar en la hoja de respuestas.

- 1** Se han admitido 2 soluciones posibles En la práctica se plantea un sistema que tolera fallos de los workers
- a** Si optamos por una solución con latidos (Heartbeats), se deben añadir sockets adicionales o cambiar el tipo de socket de los workers.
 - b** Al usar `setTimeout` sobre la petición a un worker, siempre detectamos si un worker ha caído.
 - c** Asumiendo la solución de `setTimeout`: si llega la respuesta del worker después del tiempo establecido, pero antes que la asignación a otro worker, se descarta la respuesta.
 - d** En la solución basada en latidos (Heartbeats), el broker necesita mantener un vector con los workers disponibles en un momento dado para saber si están vivos
- 2** En la solución al chat usando `zmq`:
- a** Se utilizan sockets `rep-req` entre todos los clientes para mantener actualizados los mensajes del chat
 - b** Los clientes se conocen entre ellos a través del server, de manera que se pueden mantener conversaciones privadas
 - c** Se usan un par de sockets `pull` y `pub` en el server del chat
 - d** El primer mensaje que envía el cliente es el primer mensaje introducido por teclado y es enviado a todos los clientes
- 3** Si deseamos implementar una variación del broker donde las peticiones se hagan según un tipo de trabajo o petición.
- a** El worker deberá descartar aquellos trabajos que no sean de su tipo.
 - b** El worker deberá informar al cliente del tipo de trabajo que resuelve
 - c** El broker deberá emparejar las peticiones de los clientes con los workers que resuelven ese tipo de trabajo
 - d** El broker deberá emparejar las respuestas de los workers con las peticiones de los clientes de ese tipo de trabajo
- 4** En la solución al problema planteado en prácticas para un sistema con workers con capacidad para distintos tipos de trabajo:
- a** Los workers indican en cada mensaje de respuesta el tipo de trabajo que aceptan
 - b** La petición se envía a un worker y si no es de su tipo éste responde con un mensaje específico
 - c** El broker tiene varias listas para los distintos tipos de trabajo
 - d** Si no hay un worker disponible para un tipo de trabajo, el broker asigna otro cualquiera

5 *En la implementación del sistema client-broker-worker con equilibrado de carga:*

- a** El cliente selecciona el worker con menor carga
- b** El broker necesita conocer la carga del worker mientras el worker está procesando una petición.
- c** El broker necesita conocer la carga del worker cuando el worker termina de procesar una petición o se presenta como un worker nuevo.
- d** Las estadísticas del broker que se solicitan en las prácticas hacen innecesario realizar cualquier otra gestión para el equilibrado de carga

6 *En el sistema cliente/broker/worker proporcionado, el broker realiza operaciones `bind`, y clientes y workers operaciones `connect`. Si deseamos que sean clientes y workers los que realicen las operaciones `bind` ...*

- a** No se requiere ningún cambio adicional en el código
- b** El broker necesitaría conocer las URLs de todos los clientes y workers, y clientes y workers no utilizarían la URL del broker
- c** El único cambio necesario es modificar el broker para que utilice operaciones `connect`
- d** Únicamente los clientes necesitarían obtener la URL del broker como argumento en la línea de órdenes, pero los workers no necesitarían dicha URL