

TSR – Segundo Parcial

Este examen consta de 20 cuestiones de opción múltiple. En cada cuestión, solo una opción es correcta. Debes contestar en una hoja aparte. Cada respuesta correcta aporta 0.5 puntos a la calificación. Cada respuesta errónea descuenta -0.167 puntos.

TEORÍA

1. En la configuración de despliegues, esto es un ejemplo de “inyección de dependencias”:

a	Generar una excepción cuando el puerto a utilizar ya estaba asignado a otro servidor (es decir, a uno diferente y ya en funcionamiento).
b	Iniciar otra réplica servidora cuando la carga crezca; es decir, gestionar acciones de escalado horizontal.
c	Conocer dinámicamente (o incluso actualizar) cuál es el “endpoint” de otro componente del servicio que vayamos a utilizar seguidamente.
d	No desplegar ninguna instancia de algún componente del servicio.

2. ¿Cuál de las siguientes tareas NO ESTÁ relacionada con el despliegue?

a	Instalación de un componente en ciertos ordenadores.
b	Depuración de la aplicación.
c	Resolución de dependencias.
d	Actualización de los programas a una nueva versión.

3. Supongamos que hemos escrito un algoritmo AL1 para alcanzar consenso en un sistema distribuido en el que se asume un modelo de fallos arbitrarios (*bizantinos*). Entonces:

a	AL1 es inútil, pues ningún sistema real se comporta de esa manera (es decir, el modelo de fallos arbitrarios no se presenta en ningún caso real).
b	AL1 deberá replicar todos sus procesos utilizando el modelo pasivo, pues los fallos arbitrarios solo pueden gestionarse con ese modelo de replicación.
c	AL1 será más fácil de portar a un sistema real que otro algoritmo AL2 que asuma fallos de parada, porque es más difícil que un sistema real cumpla con el supuesto de AL2.
d	AL1 será uno de los algoritmos más sencillos, pues el modelo de fallos arbitrarios facilita en gran medida la gestión de fallos en un sistema.

4. El modelo de replicación activo...

a	...puede gestionar los fallos en las réplicas más fácilmente que el modelo pasivo.
b	...es incapaz de gestionar los fallos arbitrarios.
c	...no puede proporcionar consistencia secuencial.
d	...es utilizado por omisión en los conjuntos de réplicas de MongoDB.

TSR – Segundo Parcial

5. Si se consideran las restricciones que impone el teorema CAP, cuando gestionamos particiones de la red utilizando el modelo de partición primaria...

a	...se puede proporcionar consistencia secuencial en los subgrupos de procesos minoritarios.
b	...se puede proporcionar consistencia secuencial en aquel subgrupo con una mayoría de procesos.
c	...se garantiza la disponibilidad y la consistencia entre réplicas se relaja.
d	...se renuncia a la disponibilidad en aquel subgrupo con una mayoría de procesos.

6. Si se consideran las restricciones que impone el teorema CAP, cuando debemos garantizar la disponibilidad de un servicio desplegado en varios centros de cómputo soportando el particionado de la red, entonces...

a	...podremos utilizar consistencia secuencial entre todas las réplicas del servicio.
b	...tendremos que utilizar un modelo de consistencia relajada; por ejemplo, la consistencia FIFO.
c	...no habrá ninguna restricción que afecte a la consistencia entre réplicas.
d	...la disponibilidad del servicio no se podrá garantizar en un escenario como este, compuesto por varios centros de cómputo distribuidos geográficamente.

7. ¿Cuál de estos servicios es más elástico?

a	Uno que añade una réplica cada hora, independientemente de la carga soportada en cada momento.
b	Uno que reporta periódicamente la carga actual, la capacidad de servicio y el número de réplicas al administrador, quien decide qué acción de escalado debe aplicarse.
c	Uno que adapta su número de réplicas a la carga actual de manera autónoma, manteniendo una calidad de servicio adecuada (según su SLA).
d	Uno que jamás monitorice el uso de recursos.

8. ¿Cuál de estas aproximaciones NO MEJORA la escalabilidad de un servicio?

a	Distribución de tareas.
b	Distribución de datos.
c	La toma de decisiones mediante votación entre todos los procesos participantes.
d	Uso de cachés.

TSR – Segundo Parcial

SEMINARIOS

9. ¿Qué característica distingue a las máquinas virtuales de los contenedores Docker?

a	Algunas aplicaciones a ejecutar en ellas pueden utilizar bibliotecas que no están presentes en el sistema operativo anfitrión.
b	La necesidad de un sistema operativo anfitrión.
c	El aislamiento entre contenedores y/o máquinas virtuales.
d	Las virtuales pueden ejecutar aplicaciones que no podrían ser ejecutadas en el sistema operativo anfitrión.

10. Con la orden “docker commit” se puede...

a	Finalizar una transacción iniciada para descargar o subir una imagen Docker a un almacén (“repository”).
b	Crear una imagen Docker, tomando un Dockerfile como base.
c	Iniciar un contenedor Docker, tomando una imagen Docker como base.
d	Crear una imagen Docker, a partir de un contenedor Docker.

11. Asumamos el siguiente Dockerfile, cuyo uso no generará ningún error durante su procesamiento:

```
FROM zmq  
COPY ./myProgram.js server.js  
EXPOSE 8000 8001  
CMD node server.js
```

¿Cuál de las siguientes afirmaciones ES FALSA?

a	Existe una imagen Docker “zmq”, en el almacén local o en el global.
b	Los componentes resultantes tendrán un proceso “server” que puede escuchar en el puerto 8001 del contenedor. Ese puerto corresponde al puerto 8000 en el anfitrión.
c	La imagen generada con este Dockerfile podrá ejecutar otros programas, aparte de “node server.js”.
d	Cuando se genere una imagen con este Dockerfile, debe haber un fichero llamado “myProgram.js” en la misma carpeta que el fichero Dockerfile.

TSR – Segundo Parcial

12. ¿Qué afirmación ES FALSA sobre la orden “docker-compose”?

a	Se asume que habrá un fichero “docker-compose.yml” en la carpeta donde se ejecute esa orden.
b	Siempre necesita que haya un Dockerfile para cada uno de los componentes a ejecutar.
c	Puede utilizarse para controlar un servicio basado en múltiples componentes. Las instancias de todos los componentes se ejecutan en un mismo anfitrión.
d	Su acción “scale” puede utilizarse para modificar el número de instancias de un componente del servicio.

13. ¿Qué afirmación ES FALSA sobre los modelos de consistencia “rápidos”?

a	Cuando un proceso ejecuta una acción de lectura o escritura, esa acción puede devolver el control sin haber intercambiado ningún mensaje con otros procesos.
b	El modelo de consistencia FIFO es rápido.
c	El modelo de consistencia causal es rápido.
d	El modelo de consistencia secuencial es rápido.

14. Para implantar el modelo de consistencia FIFO (utilizando ZeroMQ sobre TCP), se necesita:

a	Un proceso secuenciador que implante un orden total.
b	Que cada proceso difunda sus escrituras con un socket PUB y reciba las escrituras de los demás utilizando un único socket SUB.
c	Que cada proceso difunda sus escrituras con múltiples sockets PUB (uno por variable) y un único socket SUB para recibir las escrituras de los demás procesos.
d	Utilizar relojes vectoriales y añadir el reloj vectorial actual a cada valor escrito cuando sea difundido a los demás procesos.

15. El módulo “cluster” en NodeJS...

a	...permite que un conjunto de contenedores sea desplegado sobre varios anfitriones.
b	...amplía las órdenes “docker-compose” para desplegar programas NodeJS más rápido que sin el módulo “cluster”.
c	...permite que los programas NodeJS sean multi-hilo.
d	...permite que un conjunto de procesos trabajadores NodeJS compartan los puertos asignados a un proceso <i>master</i> NodeJS.

TSR – Segundo Parcial

16. Estamos escribiendo un programa NodeJS con el módulo “cluster”. Utiliza tantos trabajadores como procesadores haya. El proceso maestro informará al usuario cada segundo sobre cuántas peticiones han sido atendidas hasta el momento por los trabajadores.

```
var cluster = require('cluster');
var http = require('http');
if (cluster.isMaster) {
    var numReqs = 0;
    setInterval(function() { console.log("numReqs =", numReqs); }, 1000);
    function messageHandler(msg) {
        numReqs++;
    }
    var numCPUs = require('os').cpus().length;
    for (var i=0; i < numCPUs; i++) cluster.fork();
    /* (1) Worker message management instructions should be here. */
    /* (2) Worker regeneration code should be here, if any.      */
} else {
    http.Server(function(req, res) {
        res.writeHead(200); res.end('hello world\n');
        process.send({ cmd: 'notify' });
    }).listen(8000);
}
```

Elige qué instrucciones necesita el proceso maestro para gestionar (1) los mensajes de los trabajadores:

a	<code>cluster.on('message', messageHandler);</code>
b	<code>for (var i in cluster.workers) cluster.workers[i].on('message', messageHandler);</code>
c	<code>for (var i in workers) workers[i].on('notify', messageHandler);</code>
d	<code>cluster.on('notify', messageHandler);</code>

17. En el programa anterior deseamos mantener el número de trabajadores constante. Para ello, si uno muere deberá ser reemplazado por un nuevo trabajador. ¿Qué instrucciones (2) se necesitan para implantar esa funcionalidad?

a	<code>cluster.on('exit', function(worker, code, signal) { cluster.fork(); });</code>
b	<code>cluster.on('exit', function(worker, code, signal) { worker.restart(); });</code>
c	<code>cluster.on('exit', function(worker, code, signal) { worker.process.fork(); });</code>
d	<code>cluster.on('exit', function(worker, code, signal) { worker.process.start(); });</code>

TSR – Segundo Parcial

18. MongoDB utiliza los siguientes mecanismos de escalabilidad:

a	Replicación activa (o de máquina de estados) y distribución de tareas.
b	Replicación pasiva y distribución de datos (particionado horizontal).
c	Replicación activa y distribución de datos (particionado horizontal).
d	MapReduce.

19. ¿Qué afirmación ES FALSA sobre los procesos “mongos”?

a	Redirigen las peticiones que reciben.
b	Mantienen una caché de los metadatos de configuración de <i>sharding</i> de MongoDB.
c	Están ubicados en los ordenadores clientes.
d	Cada uno puede ser reemplazado por un “conjunto de réplicas”.

20. ¿Qué afirmación ES FALSA sobre el “write concern” de MongoDB?

a	Especifica cuántas réplicas deben persistir las modificaciones causadas por una operación de escritura, borrado o modificación.
b	Valores bajos en el “write concern” reducen el tiempo de servicio de las escrituras, pero ponen en peligro la persistencia de los datos en caso de fallo.
c	Su valor recomendado es “majority”, pues proporciona un buen compromiso entre rendimiento y tolerancia a fallos.
d	Necesita una confirmación por parte del cliente para cada operación solicitada.