

# TSR: Segon Parcial

Aquest examen consta de 20 qüestions d'opció múltiple. En cadascuna, només una resposta és correcta. Ha de respondre's en una altra fulla. Les respostes correctes aporten 0.5 punts a la qualificació de l'examen. Les errònies descompten 0.167 punts.

## TEORIA

1. El desplegament inclou la instal·lació inicial i la configuració d'una aplicació. A més d'això, el desplegament d'un servei també comprèn...

<b>A</b>	La depuració dels programes.
<b>B</b>	La gestió del cicle de vida del servei.
<b>C</b>	El desenvolupament de l'aplicació.
<b>D</b>	El disseny de l'aplicació.

2. L'objectiu principal de la injecció de dependències és...

<b>A</b>	Resoldre les dependències entre components utilitzant fitxers de configuració.
<b>B</b>	Eliminar totes les dependències entre components durant l'etapa de disseny.
<b>C</b>	Que la resolució de dependències siga el més transparent possible per al desenvolupador dels components.
<b>D</b>	Evitar l'ús de contenidors ja que aquests penalitzen el rendiment.

3. Imaginem un servei que necessita 400 ms per a processar localment cada petició que modifiqui el seu estat. Aquestes peticions inverteixen 20 ms per a transmetre a altres rèpliques l'estat modificat i 30 ms a aplicar les modificacions en elles. Un missatge de petició pot ser difós (en ordre total) en aquesta xarxa en 3 ms. Una petició de lectura pot ser gestionada localment en 20 ms. La proporció d'accessos és: 80% accessos de lectura i 20% accessos de modificació.

Per a escalar aquest servei, la millor aproximació serà...

<b>A</b>	Replicar-lo utilitzant el model actiu.
<b>B</b>	Replicar-lo utilitzant el model passiu, processant totes les sol·licituds en la rèplica primària.
<b>C</b>	Replicar-lo utilitzant el model passiu, però permetent que les sol·licituds de lectura siguin processades per les rèpliques secundàries.
<b>D</b>	No replicar-lo perquè la replicació introdueix massa coordinació i això impedeix un escalat eficient.

# TSR

4. En els models de consistència centrats en dades, podem dir que el model A és més fort que el model B en els següents casos:

<b>A</b>	A: causal, B: "cache".
<b>B</b>	A: FIFO, B: "cache".
<b>C</b>	A: causal, B: seqüencial.
<b>D</b>	A: causal, B: FIFO.

5. Els magatzems escalables NoSQL no suporten el model relacional perquè...

<b>A</b>	El model relacional no admet replicació.
<b>B</b>	El model relacional no admet particionat horitzontal ("sharding").
<b>C</b>	Les dades relacionals han de mantenir-se en disc.
<b>D</b>	Les transaccions en el model relacional necessiten mecanismes de control de concurrència que poden ser complexos.

6. El teorema CAP...

<b>A</b>	...exigeix que els serveis escalables i disponibles utilitzen sempre el model de consistència estricte per a tolerar així les particions de la xarxa.
<b>B</b>	...permet que els serveis escalables relaxen la seua consistència mentre la xarxa romanga particionada, assegurant així la seua disponibilitat.
<b>C</b>	...no permet la implantació de serveis altament disponibles utilitzant models de consistència forts.
<b>D</b>	...no té sentit en els centres de dades de computació en el núvol perquè no hi haurà mai particions de la xarxa en ells.

7. Respecte a l'escalabilitat de serveis es pot afirmar que...

<b>A</b>	Un mateix servei no pot escalar horitzontalment i vertical.
<b>B</b>	Els algorismes descentralitzats milloren l'escalabilitat de distància.
<b>C</b>	El particionat horitzontal ("sharding") millora l'escalabilitat administrativa.
<b>D</b>	Evitar la contenció és un factor clau per a millorar l'escalabilitat d'un servei.

# TSR

## 8. Els objectius principals d'un subsistema de seguretat són:

<b>A</b>	Protecció, control d'accés i seguretat física.
<b>B</b>	Protecció, gestió de la confiança i un bon suport per a mecanismes de xifrat.
<b>C</b>	Comptabilitat, integritat, confidencialitat i disponibilitat.
<b>D</b>	Polítiques robustes, mecanismes eficients i garanties correctes.

## SEMINARIS

### 9. Quina de les següents tasques NO es realitza en utilitzar aquesta ordre Docker? `docker run -it ubuntu /bin/bash`

<b>A</b>	Executar el programa <code>/bin/bash</code> en un contenidor.
<b>B</b>	Descarregar la imatge "ubuntu:latest" des de Docker Hub si no la teníem en el dipòsit d'imatges local.
<b>C</b>	Arreplegar l'eixida del contenidor que està sent utilitzat per a executar aquesta ordre. L'eixida podrà mostrar-se mitjançant l'ordre <code>docker logs</code> .
<b>D</b>	Eliminar automàticament aquest contenidor una vegada la seua execució haja acabat.

### 10. L'ordre `docker commit a b`...

<b>A</b>	Crea un nou contenidor anomenat "a" utilitzant el Dockerfile situat en la carpeta "b".
<b>B</b>	Crea una nova imatge "b" utilitzant el Dockerfile de la carpeta "a".
<b>C</b>	Crea una nova imatge "b" amb el contingut actual del contenidor el nom o identificador del qual és "a".
<b>D</b>	Realitza el "commit" d'una transacció "a" que va ser iniciada amb una ordre <code>docker pull</code> o <code>docker push</code> , generant un contenidor amb ID "b".

### 11. El mòdul "cluster" de NodeJS s'utilitza per a...

<b>A</b>	...desplegar un conjunt de programes NodeJS en un "cluster" d'ordinadors.
<b>B</b>	...gestionar múltiples fils en un procés NodeJS.
<b>C</b>	...gestionar un conjunt de processos NodeJS perquè puguin compartir alguns recursos; p. ex., un port en el qual escoltar i un mateix programa a executar.
<b>D</b>	...implantar fàcilment múltiples models de consistència de memòria.

# TSR

## 12. MongoDB utilitza els següents mecanismes per a millorar la seua escalabilitat:

<b>A</b>	Control de concurrència distribuït.
<b>B</b>	Algorismes descentralitzats.
<b>C</b>	Replicació passiva i particionat horitzontal (o “ <i>sharding</i> ”).
<b>D</b>	Escalabilitat administrativa.

## 13. L'objectiu principal dels servidors de configuració en MongoDB és:

<b>A</b>	Adoptar el paper d'àrbitres quan una rèplica falle.
<b>B</b>	Controlar la distribució de dades entre les múltiples particions horitzontals (“shards”) existents.
<b>C</b>	Respectar el teorema CAP quan es done una partició en la xarxa.
<b>D</b>	Detectar fallades en els nodes, iniciant un protocol de recuperació quan es done alguna fallada.

## 14. Considerant la classificació temàtica de vulnerabilitats vista en el Seminari 8, es pot considerar certa la següent afirmació:

<b>A</b>	L'explotació de defectes en polítiques de seguretat no pot automatitzar-se tan fàcilment com l'explotació de contrasenyes febles.
<b>B</b>	El “ <i>phishing</i> ” és una vulnerabilitat de tipus “error de programari”.
<b>C</b>	La protecció física és un exemple de vulnerabilitat d'enginyeria social.
<b>D</b>	Un defecte en una política de seguretat de protecció personal requereix menor interacció per a ser explotada que un error de programari en el sistema operatiu.

## 15. Assumint aquest Dockerfile...

```
FROM zmq
RUN mkdir /zmq
COPY ./broker.js /zmq/broker.js
WORKDIR /zmq
EXPOSE 8000 8001
CMD node broker.js
```

### Quina de les següents afirmacions és **FALSA**?

<b>A</b>	Necessitem tenir el fitxer “broker.js” en el directori de l'amfitrió en el qual es trobe aquest Dockerfile.
<b>B</b>	El programa a executar en aquests contenidors utilitza el port 8000 del contenidor i l'associa al port 8001 de l'amfitrió.
<b>C</b>	Per omissió, els contenidors generats a partir d'aquest Dockerfile executaran l'ordre “ <b>node broker.js</b> ”.
<b>D</b>	Aquest Dockerfile assumeix l'existència d'una imatge anomenada “zmq” amb una instal·lació vàlida de l'interpret de JavaScript “node”.

# TSR

16. Imaginem que el component broker de la qüestió 15 s'inclou en un fitxer docker-compose.yml amb aquests continguts (entre uns altres que corresponguen a altres components):

```
version: '2'
services:
  ...
  bro:
    image: broker
    build: ../broker/
```

Quina de les següents afirmacions és FALSA?

<b>A</b>	Podem iniciar almenys una instància del component <b>broker</b> amb l'ordre <b>docker-compose up -d</b>
<b>B</b>	Una vegada s'ha iniciat el servei, podem tenir 5 instàncies del component <b>broker</b> utilitzant <b>docker-compose scale broker=5</b>
<b>C</b>	Aquest fitxer "docker-compose.yml" assumeix que el Dockerfile mostrat en la qüestió 15 es troba en la carpeta "../broker/".
<b>D</b>	Podem utilitzar l'ordre <b>docker-compose stop bro</b> per a parar totes les instàncies d'aquest component <b>broker</b> .

17. Suposem un protocol de replicació basat en un procés seqüenciador que utilitza un socket PUB ZeroMQ per a propagar tots els esdeveniments "write" als processos participants, en ordre de recepció. Aquests esdeveniments han sigut rebuts mitjançant canals PUSH-PULL, on el seu socket PULL està en el seqüenciador. Aquest protocol de replicació suporta els següents models de consistència:

<b>A</b>	Només el model estricte.
<b>B</b>	Només el model "cache".
<b>C</b>	Només el model causal.
<b>D</b>	Seqüencial, processador, causal, "cache" i FIFO.

18. Aquesta execució...

W1(x)1, R4(x)1, W2(y)2, W1(y)3, W2(x)4, R3(y)2, W3(x)5, R1(x)5, R2(x)1, R3(x)1, R4(y)3, R1(y)2, R3(y)3, R4(x)5, R3(x)4, R2(y)3, R4(y)2, R1(x)4, R2(x)5, R4(x)4.

...suporta aquests models de consistència:

<b>A</b>	Només el model FIFO.
<b>B</b>	Només el model "cache".
<b>C</b>	Processador, FIFO i "cache".
<b>D</b>	Seqüencial, processador, causal, "cache" i FIFO.

# TSR

19. Donat el següent programa servidor de descàrrega de fitxers...

<pre>var cluster = require('cluster'); var fs = require('fs'); var path = require('path'); var zmq = require('zmq'); var os = require('os'); const ipcName = 'Act2.ipc'; const dlName = 'ipc://' + ipcName; if (cluster.isMaster) {   var numCPUs = os.cpus().length;   var rt = zmq.socket('router');   var dl = zmq.socket('dealer');   rt.bindSync('tcp://127.0.0.1:8000');   dl.bindSync(dlName);   rt.on('message', function() {     msg = Array.apply(null, arguments);     dl.send(msg); });   dl.on('message', function() {     msg = Array.apply(null, arguments);     rt.send(msg); }); }</pre>	<pre>} else {   var rep = zmq.socket('rep');   rep.connect(dlName);   rep.on('message', function(data) {     var request = JSON.parse(data);     fs.readFile(request.path, function(err,     data) {       if (err) data = ' NOT FOUND';       rep.send(JSON.stringify({         pid: process.pid ,         path: request.path ,         data: data ,         timestamp: new Date().toString()       })))     })   }) }</pre>
---	---

Hem tractat d'executar el programa, però no sembla fer res útil. El seu principal problema és...

<b>A</b>	Els sockets ZeroMQ no admeten “ipc://” com a transport.
<b>B</b>	No s'ha creat cap procés treballador del mòdul “cluster”.
<b>C</b>	S'està tractant de propagar missatges interns del mòdul “cluster” a través d'un socket DEALER ZeroMQ.
<b>D</b>	Un servidor no pot utilitzar un socket ROUTER com el seu “endpoint”.

20. La vulnerabilitat OpenSSL Heartbleed descrita en el Seminari 8 és un exemple de vulnerabilitat que pertany a les classes següents:

<b>A</b>	“Defecte en la política de treball” en la seua categoria i “Personal humà” en l'origen.
<b>B</b>	“Defecte en la lògica del programari” en la seua categoria i “Biblioteca/middleware” en l'origen.
<b>C</b>	“Enginyeria social” en la seua categoria i “Desenvolupador” en l'origen.
<b>D</b>	“Defecte en la lògica del programari” en la seua categoria i “Personal humà” en l'origen.