

Aquest examen té un valor de 10 punts, i consta de 32 qüestions. Cada qüestió planteja 4 alternatives i té una única resposta correcta. Cada resposta correcta aporta 10/32 punts, y cada error descompta 10/96 punts. Ha de contestar en la fulla de respostes.

Hem desenvolupat el següent codi per a generar un sistema al qual anomenarem ABC:

```
// client.js
const zmq = require("zeromq")
const url = process.argv[2]
let push = zmq.socket('push')
push.connect(url)
push.send('hola 1')
push.send('hola 2')
push.send('hola 3')
```

```
// a.js
const zmq = require("zeromq")
const [port1,port2]= process.argv.slice(2)
let pull = zmq.socket('pull')
let push = zmq.socket('push')
pull.bindSync('tcp://*:'+port1)
push.bindSync('tcp://*:'+port2)
pull.on('message', (m) => {push.send(m)})
```

```
// b.js
const zmq = require("zeromq")
const [url1,url2]= process.argv.slice(2)
let pull = zmq.socket('pull')
let push = zmq.socket('push')
pull.connect(url1)
push.connect(url2)
pull.on('message', (m) => {push.send(m)})
```

```
// c.js
const zmq = require("zeromq")
const port = process.argv[2]
let pull = zmq.socket('pull')
pull.bindSync('tcp://*:'+port)
pull.on('message', (m) => {console.log(m+"")})
```

Realitzem un desplegament manual, llançant les següents ordres en terminals diferents

```
node c 8002
node a 8000 8001
node b tcp://127.0.0.1:8001 tcp://127.0.0.1:8002
node b tcp://127.0.0.1:8001 tcp://127.0.0.1:8002
node client tcp://127.0.0.1:8000
```

Veiem que en el primer terminal (el de l'ordre node c 8002) el programa c escriu

```
hola 1
hola 2
hola 3
```

Volem automatitzar el desplegament usant contenidors de manera que els clients accedisquen de forma remota a través del port 80 de l'amfitrió. Per a fer això hem preparat una imatge base denominada *tsr2021/ubuntu-zmq* (amb la distribució *ubuntu* i suport per a *nodeJS* i *ZeroMQ*) i la següent estructura de directoris:

```
p2/docker-compose.yml
p2/a/Dockerfile
p2/a/a.js
p2/b/Dockerfile
p2/b/b.js
p2/c/Dockerfile
p2/c/c.js
```

En el *Dockerfile* per a *a.js* hi haurà una línia amb aquest contingut:

```
CMD node a 8000 8001
```

D'altra banda, al *Dockerfile* per al programa *c.js* tindrem aquesta instrucció:

```
CMD node c 8002
```

Assumint aquest escenari, EN EL QUAL EL CLIENT NO ÉS UN ELEMENT A CONSIDERAR DURANT EL DESPLEGAMENT AUTOMATITZAT, responga a les següents qüestions:

1 *En relació amb el desplegament manual del sistema ABC:*

- a** Apareixen 5 components diferents, amb una instància cadascun
- b** Apareixen 4 components diferents, un d'ells amb 2 instàncies
- c** Usem un únic component, ja que despleguem tots els programes en la mateixa màquina
- d** Usem un únic component, ja que tots els programes utilitzen el mateix suport per a execució (nodeJS + ZMQ)

2 *En el sistema ABC AMB DESPLEGAMENT AUTOMATITZAT:*

- a** No apareixen dependències
- b** Únicament hi ha dependències dels clients respecte d'a
- c** Hi ha dependències de 'b' respecte d'a i 'c', i de clients respecte d'a
- d** Únicament hi ha dependències de 'b' respecte d'a i 'c'

3 *Quant a l'automatització del desplegament del sistema ABC per a executar 'a', 'b' i 'c' en la mateixa màquina:*

- a** L'ús de contenidors no és adequat, ja que dins d'una mateixa màquina l'única alternativa és usar màquines virtuals
- b** Les parts 'b' i 'c' poden utilitzar contenidors, però la part 'a' ha d'atendre missatges de clients remots, i per tant únicament pot implementar-se mitjançant una màquina virtual
- c** Podem usar contenidors per a les parts 'a', 'b' i 'c'
- d** Si llancem 'a', 'b', 'c' en la mateixa màquina, no podem llançar clients de forma remota: han de llançar-se també en l'amfitrió

4 *El fitxer p2/b/Dockerfile del sistema ABC té el següent contingut*

```
FROM tsr2021/ubuntu-zmq
COPY ./b.js b.js
CMD node b ...
```

Indica què hem d'afegir en els punts suspensius de la tercera línia

- a** tcp://127.0.0.1:8001 tcp://127.0.0.1:8002
- b** tcp://127.0.0.1:8001 \$URL
- c** \$URL tcp://127.0.0.1:8002
- d** \$URL1 \$URL2

5 *Assumint que el fitxer p2/b/Dockerfile del sistema ABC té el contingut correcte i que estem situats en el directori p2/, per a crear de forma automàtica la imatge corresponent a 'b' (que anomenarem imb) hem d'executar:*

- a** docker build -t imb .
- b** docker commit -t imb .
- c** docker build -t imb ./b
- d** docker run -t imb .

6 *Assumint que els punts suspensius podran representar, si fóra necessari, la llista d'arguments o valors per a les variables d'entorn a utilitzar pel programa 'b' del sistema ABC i que s'ha creat correctament la imatge imb corresponent a aqueix programa, indica l'ordre necessària per a llançar una instància de la mateixa sobre un contenidor*

- a** docker build imb ...
- b** docker commit imb ...
- c** docker start imb ...
- d** docker run imb ...

7 *En el fitxer p2/a/Dockerfile del sistema ABC convé utilitzar la següent línia*

- a** EXPOSE 8000 8001
- b** PORTS 8000 8001
- c** Tant EXPOSE 8000 8001 com PORTS 8000 8001
- d** Ni EXPOSE 8000 80001 ni PORTS 8000 8001

- 8** *En relació amb el servei 'a' del fitxer docker-compose.yml del sistema ABC*
- a** Requereix part 'links' i part 'environment'
 - b** Requereix part 'links', però no 'environment'
 - c** Requereix part 'environment', però no 'links'
 - d** No requereix part 'environment' ni part 'links'
- 9** *En relació amb el servei 'a' del fitxer docker-compose.yml del sistema ABC:*
- a** Requereix part 'ports'.
 - b** La resta d'alternatives són incorrectes.
 - c** Requereix part 'expose', però no 'ports'.
 - d** No requereix part 'ports' ni part 'expose'.
- 10** *En el fitxer docker-compose.yml del sistema ABC:*
- a** Únicament necessita una secció 'environment' el servei 'b'
 - b** Únicament necessiten una secció 'environment' els serveis 'a' i 'c'
 - c** Únicament necessita una secció 'environment' el servei 'a'
 - d** Cap servei necessita una secció 'environment'
- 11** *Si en el sistema ABC tots els fitxers són correctes, una ordre correcta per a llançar una instància d'a, una de 'b' i una de 'c' és*
- a** `docker run ima & docker run imb & docker run imc &`
 - b** `docker-compose up`
 - c** `docker build --scale ima=1 --scale imb=1 --scale imc=1`
 - d** No es pot completar amb una única ordre: cal realitzar part del treball a mà (esbrinar la ip del contenidor, editar els Dockerfiles, etc.)
- 12** *Si en el sistema ABC hem desplegat correctament els programes 'a', 'b' i 'c' de forma automàtica, la forma correcta de llançar un client en la mateixa màquina és:*
- a** Executant des d'un altre terminal l'ordre
`node client tcp://127.0.0.1:80`
 - b** Executant des d'un altre terminal l'ordre
`node client tcp://127.0.0.1:8000`
 - c** No es pot llançar el client en la mateixa màquina (ha de ser remot)
 - d** Executant l'ordre
`docker run client tcp://127.0.0.1:80`
- 13** *En relació amb la interoperabilitat de sockets DEALERS i REPs quin dels següents enunciats és FALS?*
- a** Tot missatge enviat per un DEALER a un REP ha d'incloure, almenys, un segment buit.
 - b** El DEALER ha d'operar de manera sincrònica ja que el REP així ho requereix.
 - c** Diversos DEALERS poden estar connectats i operant amb diversos REPs.
 - d** Un REP pot estar connectat i operant amb un conjunt de sockets en el qual hi hagen tant DEALERS com REQs.

- 14** Es modifica el sistema clients-broker-workers descrit en teoria (vegeu, per exemple, els codis del tema 4) de la manera següent:

1) El broker en lloc de disposar de dos sockets de tipus ROUTER disposa de només un que presta servei tant als clients com als workers.

2) En conseqüència els clients i els workers han sigut modificats de manera que comparteixen el mateix endpoint (del ROUTER) i a més són anònims, això és, no s'ha assignat cap identificador al seu atribut identity.

Tria l'afirmació correcta:

- a** El codi del broker original pot ser transformat (amb modificacions immediates) per a obtenir el d'aquest nou broker de manera que tot el sistema pugui operar de la manera habitual.
- b** Aquesta modificació no es pot dur a terme ja que el broker no podria distingir entre els clients i els workers.
- c** Aquesta modificació només es podria implantar si els workers usaren sockets de tipus REP.
- d** Cap dels restants enunciats és vàlid.

- 15** Selecciona quin dels següents enunciats és FALS:

- a** Els missatges enviats pels sockets DEALER sempre han d'incloure algun segment buit.
- b** Els sockets ROUTER són asincrònics.
- c** Els sockets ROUTER utilitzen identitats per a gestionar els missatges que envien i reben.
- d** Els sockets DEALER poden interoperar entre ells.

- 16** Selecciona quin dels següents enunciats és FALS:

- a** Els sockets DEALER són sincrònics.
- b** Els sockets DEALER encuen els missatges que encara no han pogut enviar.
- c** Els sockets DEALER encuen els missatges rebuts encara no processats.
- d** Els sockets DEALER poden interoperar amb sockets del tipus ROUTER.

- 17** La següent traça, on:

- un element de la forma $1xW2$ indica que el procés (1) aplica sobre la variable (x) l'operació d'escriptura (W) generant el valor (2)

- un element de la forma $3yR1$ indica que el procés (3) aplica sobre la variable (y) l'operació de lectura (R), obtenint el valor (1)

1xW1	2xR1	4xR1	3xR1	2xW2	1xW3	3xR3	4xR2	3xR2	4xR3
------	------	------	------	------	------	------	------	------	------

- a** Compleix consistència estricta
- b** Compleix consistència seqüencial
- c** Compleix consistència causal
- d** Totes les altres afirmacions són falses

- 18** Si una determinada traça compleix consistència seqüencial

- a** No podem assegurar que complisca consistència processador
- b** No podem assegurar que complisca consistència causal
- c** No podem assegurar que complisca consistència 'cache'
- d** Totes les altres afirmacions són falses

- 19** En un sistema amb operacions no deterministes i sense errors bizantins, i amb duració de cada operació 10 vegades el cost de propagar i aplicar modificacions

- a** Hem de triar replicació activa si despleguem un únic servei
- b** Hem de triar replicació passiva únicament si despleguem més d'un servei
- c** Els dos models (replicació activa / replicació passiva) són vàlids
- d** Hem de triar replicació passiva amb independència del nombre de serveis desplegats

- 20** *En un sistema amb totes les operacions deterministes i possibilitat d'errors bizantins, i amb duració de cada operació 10 vegades el cost de propagar i aplicar modificacions*
- a** Hem de triar replicació passiva
 - b** Hem de triar replicació multi-master
 - c** Hem de triar replicació activa exclusivament si despleguem un únic servei i replicació passiva si despleguem més d'un servei
 - d** Hem de triar replicació activa amb independència del nombre de serveis desplegats
- 21** *Què ocorre si en el model passiu es permet que les rèpliques secundàries responguen directament a les operacions de consulta enviades pels clients?*
- a** Es toleraran les fallades arbitràries (o bizantines).
 - b** Es toleren millor les particions en la xarxa d'intercomunicació.
 - c** No es podran gestionar de manera consistent les operacions no deterministes.
 - d** Millora el rendiment dels serveis replicats d'aqueixa manera.
- 22** *Un magatzem NoSQL és una base de dades en la qual (entre altres possibles característiques):*
- a** No s'utilitza el model relacional ni el suport transaccional.
 - b** Se simplifica l'esquema.
 - c** Es permet el particionat i el repartiment de les dades entre diferents nodes.
 - d** Totes les altres opcions són correctes.
- 23** *Segons el teorema CAP, quan ocorrega una partició de la xarxa...*
- a** Tots els subgrups continuaran responent als seus clients i els serveis replicats podran mantenir consistència seqüencial.
 - b** Els serveis replicats podran respectar una consistència causal si han de mantenir la seua disponibilitat.
 - c** S'utilitzarà un model de partició primària i totes les rèpliques del servei podran respondre als seus clients.
 - d** No es permetrà cap activitat dels serveis distribuïts en aqueix sistema durant aqueix interval.
- 24** *Quina de les següents afirmacions sobre la replicació multi-màster és CERTA?*
- a** És capaç de gestionar més models de fallades que la replicació activa.
 - b** Necessita una difusió d'ordre total per a garantir la consistència entre les seues rèpliques.
 - c** Permet que les instruccions de cada operació sol·licitada siguin executades per una sola rèplica.
 - d** Requereix que les operacions de lectura siguin contestades per totes les seues rèpliques per a suportar el model de fallades arbitràries.
- 25** *Quin tipus de replicació utilitza MongoDB?*
- a** Cap, perquè aconsegueix millorar la seua escalabilitat mitjançant particionat/distribució de dades.
 - b** Multi-màster.
 - c** Passiva mentre la xarxa no estiga particionada i multi-màster quan hi haja particions en la xarxa.
 - d** Passiva, però admet en la seua configuració que les rèpliques secundàries atenguen consultes.

- 26** *El mòdul cluster de Node.js s'utilitza per a:*
- a** Desplegar un conjunt de programes Node.js en un clúster d'ordinadors.
 - b** Gestionar múltiples fils d'execució en un procés Node.js.
 - c** Gestionar un conjunt de processos Node.js perquè puguin compartir el programa a executar i el port en el qual rebre sol·licituds.
 - d** Facilitar el desenvolupament de proxies que propaguen peticions a un clúster d'ordinadors on desplegar múltiples instàncies servidores.
- 27** *Durant el desenvolupament de la pràctica 3 es creen imatges amb el mateix nom corresponents a diferents versions del programa. Per a evitar conflictes es recorre a:*
- a** Deixar les imatges amb el mateix nom en directoris diferents
 - b** Esborrar les imatges anteriors abans de generar les noves
 - c** No hi ha possibilitat de conflicte, perquè les imatges noves substitueixen automàticament a les preexistents amb el mateix nom
 - d** No hi ha possibilitat de conflicte, perquè el sistema tria sempre la versió amb data més recent
- 28** *En la variant CBW de la pràctica 3 que accepta tipus de treballs assumim que:*
- el client té un codi únic que genera un tipus de treball (petició) o un altre segons un argument en línia d'ordres
 - el worker té un codi únic que l'habilita per a un tipus de treball o un altre segons un argument en línia d'ordres
- En aqueix escenari, i assumint que després volem controlar el nombre d'instàncies de cada tipus de client i de worker:*
- a** Podem tenir el codi dels clients en un únic directori, i el dels workers en un únic directori
 - b** Necessitem un directori diferent per tipus de worker, però per als clients és suficient un únic directori
 - c** Necessitem un directori diferent per tipus de client, però per als workers és suficient un únic directori
 - d** Necessitem un directori diferent per tipus de client, i un directori diferent per tipus de worker
- 29** *En la variant CBW de la pràctica 3 que accepta tipus de treballs assumim que:*
- el client té un codi únic que genera un tipus de treball (petició) o un altre segons un argument en línia d'ordres
 - el worker té un codi únic que l'habilita per a un tipus de treball o un altre segons un argument en línia d'ordres
- En aqueix escenari, i assumint que després volem controlar el nombre d'instàncies de cada tipus de client i de worker:*
- a** El fitxer docker-compose.yml ha d'incloure una entrada diferent per a cada tipus de client i una entrada diferent per a cada tipus de worker
 - b** El fitxer docker-compose.yml ha d'incloure una entrada diferent per a cada tipus de client, però és suficient una entrada única per als workers
 - c** El fitxer docker-compose.yml ha d'incloure una entrada diferent per a cada tipus de worker, però és suficient una entrada única per als clients
 - d** En el fitxer docker-compose.yml és suficient una entrada única per a tots els clients, i una altra entrada única per a tots els workers

- 30** En l'apartat de desplegament manual de la pràctica 3 es té per als clients el següent Dockerfile

```
FROM tsr2021/ubuntu-zmq
COPY ./2client.js myclient.js
CMD node myclient NEED_BROKER_URL
```

Siga imclient la seua imatge corresponent. Considere's el següent Dockerfile

```
FROM imclient
ENTRYPOINT ["node"]
CMD ["myclient"]
```

situat en un nou directori, on executem:

```
docker build -t nuevocliente .
```

En aquestes condicions:

- a** El nou Dockerfile no està ben definit.
- b** La imatge nuevocliente no està adequadament construïda.
- c** Amb la imatge nuevocliente poden construir-se i executar-se clients que poden interoperar amb el broker si reben la informació adequada.
- d** Encara que amb la imatge nuevocliente puguen generar-se contenidors clients, aquests no tindran manera d'interoperar amb el broker.

- 31** En la segona part de la tercera pràctica es considera l'ús d'un logger en les condicions allí presentades. Considere's la següent variant: el logger només rebrà anotacions per part del broker i s'ha especificat que en el desplegament el primer servei desplegat siga el broker. Per a fer això s'han modificat adequadament els codis del broker i del logger; només en els aspectes mínims requerits, per a considerar aquestes condicions conforme a com el següent docker-compose.yml mostra.

```
version: '2'
services:
  ...
  bro:
    image: broker
    build: ./broker/
    expose:
      - "9997"
      - "9998"
      - "9999"
  log:
    image: logger
    build: ./logger/
    links:
      - bro
    volumes:
      # /tmp/logger.log DIRECTORY must
      exist on host and be writeable
      - /tmp/logger.log:/tmp/cbwlog
    environment:
      - LOGGER_DIR=/tmp/cbwlog
      # El logger fa connect
      - BROKER_URL=tcp://bro:9997
```

En aquestes condicions:

- a** Aquest desplegament no és factible i reportarà un error.
- b** El sistema pot desplegar-se però en desplegar primer el broker el logger podria no rebre les primeres anotacions.
- c** El sistema així desplegat funcionarà adequadament.
- d** Perquè el sistema funcionara adequadament caldria desplegar el logger abans que els clients i els workers.

32 *En la tercera sessió de la pràctica 3 ha de desplegar-se un component worcli que intercomunica dos sistemes client-broker-treballador. Trie l'afirmació correcta:*

- a** Worcli ha de desplegar-se forçosament en l'amfitrió del primer broker.
- b** Worcli ha de desplegar-se forçosament en l'amfitrió del segon broker.
- c** Worcli només pot comunicar-se amb altres Worcli.
- d** Worcli no és estrictament necessari. Els brokers d'aqueixos dos sistemes, sense modificacions, podrien comunicar-se directament sense cap component intermediari.



Emplena i entrega aquest full de respostes. Cada qüestió té una única resposta correcta. No oblidis emplenar correctament les teues dades personals.

No has de fer cap marca damunt d'una possible resposta incorrecta: esborra-la o cobreix-la amb Tipp-Ex

Una qüestió amb més d'una resposta marcada es considera no contestada

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

DNI: _____

Cognoms: _____

Nom: _____

1	A	B	C	D
2	A	B	C	D
3	A	B	C	D
4	A	B	C	D
5	A	B	C	D
6	A	B	C	D
7	A	B	C	D
8	A	B	C	D
9	A	B	C	D
10	A	B	C	D
11	A	B	C	D
12	A	B	C	D
13	A	B	C	D
14	A	B	C	D
15	A	B	C	D
16	A	B	C	D
17	A	B	C	D
18	A	B	C	D
19	A	B	C	D
20	A	B	C	D

21	A	B	C	D
22	A	B	C	D
23	A	B	C	D
24	A	B	C	D
25	A	B	C	D
26	A	B	C	D
27	A	B	C	D
28	A	B	C	D
29	A	B	C	D
30	A	B	C	D
31	A	B	C	D
32	A	B	C	D

