

*Esta parte consta de tres preguntas tipo test y dos modificaciones de un programa. Las modificaciones aportan 0.4 puntos. Cada pregunta del test tiene 4 opciones y solo una de ellas es correcta. Cada respuesta correcta aporta 0.2 puntos y cada error descuenta 0.067 puntos. El test debe responderse en la hoja final. Las modificaciones deben realizarse en una hoja aparte.*

El siguiente programa NodeJS está inspirado en un ejemplo del Tema 6. Modifícalo de manera que cada 5 segundos:

- 1 Muestre cuántas peticiones han sido procesadas por todos los trabajadores en esos 5 segundos. (0.2 puntos)
- 2 Si el número de peticiones atendidas en esos 5 segundos es más de 100 veces mayor que el número actual de trabajadores, deberá crearse otro trabajador. (0.2 puntos)

```
const cluster = require('cluster')
const http = require('http')
const interval = 1000
if (cluster.isMaster) {
  let numReqs = 0
  function statistics() {
    console.log("numReqs =" + numReqs)
  }
  setInterval(statistics, interval)
  function messageHandler(msg) {
    if (msg.cmd && msg.cmd === 'notifyRequest')
      numReqs++
  }
  const numCPUs = require('os').cpus().length
  for (let i=0; i < numCPUs/2; i++) {
    let p=cluster.fork()
    p.on('message', messageHandler)
  }
} else {
  http.Server(function(req, res) {
    res.writeHead(200); res.end('hello world\n')
    process.send({ cmd: 'notifyRequest' })
  }).listen(8000)
}
```

- 1 *El programa original listado anteriormente:*
  - a No muestra ningún mensaje.
  - b Muestra periódicamente cuántas peticiones han sido servidas por el primer trabajador.
  - c Muestra periódicamente cuántas peticiones han sido atendidas por todos los trabajadores.
  - d Muestra cuántas peticiones han sido servidas por el primer trabajador durante su primer segundo de ejecución.
- 2 *En el programa original:*
  - a Solo un proceso escucha en el puerto 8000 del ordenador local.
  - b Se inician varios procesos trabajadores y cada uno usa el puerto 8000 en un ordenador diferente.
  - c Se inician varios procesos trabajadores y todos comparten el puerto 8000 del ordenador local.
  - d El proceso maestro aborta al recibir su primer mensaje, pues los mensajes no tienen ninguna propiedad cmd.
- 3 *El programa ampliado que debemos implementar:*
  - a Necesita usar el módulo 'zmq' de NodeJS.
  - b Necesita que explícitamente importemos el módulo 'events' de NodeJS.
  - c No puede escalar de ninguna manera.
  - d Es parcialmente elástico, pues puede reaccionar autónomamente a incrementos en su carga.