TSR

Examen de las sesiones 1 y 2 de la Práctica 2

1. Dado el código del fichero origen1.js de la práctica 2...

```
 1: const {zmq, lineaOrdenes, error, adios, conecta} = require('../tsr')
 2: lineaOrdenes("nombre hostSig portSig")
 3: let salida = zmq.socket('push')
 4: conecta(salida, hostSig, portSig)
 5: salida.on('error', (msg) => {error(`${msg}`)})
 6: process.on('SIGINT', adios([salida],"abortado con CTRL-C"))
 7: for (let i=1; i<=4; i++) {
 8:         console.log(`enviando mensaje: [${nombre},${i}]`)
 9:         salida.send([nombre,i])
10: }
```

Y el código del fichero destino.js:

```
 1: const {zmq, error, lineaOrdenes, traza, adios, creaPuntoConexion} = require('../tsr')
 2: lineaOrdenes("nombre port")
 3: let entrada = zmq.socket('pull')
 4: creaPuntoConexion(entrada, port)
 5: function procesaMensaje(filtro, nombre, iteracion) {
 6:         if (!iteracion) {
 7:                 iteracion = nombre
 8:                 nombre = filtro
 9:                 filtro = ""
10:         }
11:         traza('procesaMensaje','filtro nombre iteracion',[filtro, nombre, iteracion])
12: }
13: entrada.on('message', procesaMensaje)
14: entrada.on('error', (msg) => {error(`${msg}`)})
15: process.on('SIGINT', adios([entrada],"abortado con CTRL-C"))
```

Se solicita escribir el programa *filtro.js*, en el que no será necesario gestionar los posibles eventos 'error', y adaptar el programa *origen1.js* de manera que respeten todas estas condiciones simultáneamente:

- **Un solo proceso que ejecute el programa *origen1.js* adaptado podrá interactuar con tantos procesos que ejecuten *filtro.js* como decida el usuario. (15%)**
- **El programa *origen1.js* adaptado no debe enviar cuatro mensajes, sino un mensaje cada segundo, mientras dure su ejecución. El primer mensaje incluirá un 1 como valor de su segundo segmento. Cada nuevo envío incrementará ese valor en una unidad. (20%)**
- **El programa *filtro.js* recibirá desde la línea de órdenes el nombre de ese filtro (el usuario podrá iniciar varios, cada uno con un nombre diferente), así como la información necesaria para intercomunicarse con *origen1.js* y *destino.js*. (15%)**
- **Al filtrar cada mensaje recibido, el programa *filtro.js* incluirá su nombre como primer segmento adicional y duplicará el valor recibido en el último segmento. (15%)**
- **El programa *filtro.js* debe utilizar los tipos de socket adecuados y las operaciones necesarias para que la comunicación sea posible entre los tres tipos de proceso. (15%)**
- **El programa *filtro.js* espera al menos el número de milisegundos especificado en el último segmento del mensaje recibido antes de reenviar ese mensaje filtrado a *destino.js*. (20%)**