

# NIST

## Exam of Sessions 1 and 2 of Practice 2

1. Given the following code from file *origen1.js* of practice 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: }
```

And the following code from file *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"))
```

Write a program *filtro.js*, in which it will not be necessary to manage possible 'error' events, and adapt *origen1.js* so that the following conditions are all met at the same time:

- A single process running the adapted program *origen1.js* will be able to interact with as many processes running *filtro.js* as the user decides. (15%)
- The adapted program *origen1.js* should not send four messages, but one message per second while it is running. The first message will include a 1 as the value of its second segment. Each new message will increment this value by one unit. (20%)
- Program *filtro.js* will receive the name of the filter (the user will be able to start several, each one with a different name) from the command line, as well as the necessary information to communicate with *origen1.js* and *destino.js*. (15%)
- When filtering each received message, program *filtro.js* will include its name as the first additional segment and will double the value received in the last segment. (15%)
- Program *filtro.js* must use the proper socket types and the necessary operations so that communication will be possible between all three kinds of processes. (15%)
- Program *filtro.js* should wait at least the number of milliseconds specified in the last segment of the received message before forwarding that filtered message to *destino.js*. (20%)