TSR	Recuperación	Práctica 1	2019-01-25	A

Esta prueba tiene un valor de 1 punto, y consta de 6 cuestiones tipo test. Cada cuestión plantea 4 alternativas y tiene una única respuesta correcta. Cada respuesta correcta aporta 1/6 puntos, y cada error descuenta 1/18 puntos. Debe contestar en la hoja de respuestas.

1 *Señala la opción correcta sobre el proxy desarrollado en el bloque 1 de prácticas:*

a La respuesta del servidor remoto accedido a través del proxy llega directamente al cliente.

b Todas las respuestas del servidor remoto accedido a través del proxy son procesadas por el mismo listener en el proxy.

c Cada vez que se conecta un cliente al proxy, un nuevo listener se asigna y crea una nueva conexión al servidor remoto.

d Las respuestas del servidor remoto se reciben en el proxy en el mismo orden en el que recibe las peticiones el proxy efectuadas desde distintos clientes.

2 *La funcion* `getLoad()`*, usada en el bloque 1 de prácticas*

a Determina la capacidad del servidor para recibir peticiones.

b Devuelve la cantidad de peticiones que están siendo procesadas por el servidor.

c Requiere la IP del cliente para poder responder al mismo.

d Devuelve junto al resultado la IP del servidor, para determinar claramente el servidor al que corresponde el resultado.

3 *Suponemos el módulo* `generadorEventos.js` *de la práctica 1 (primer código), y una aplicación que crea dos objetos de tipo Evento y emite cíclicamente los eventos 1 y 2 por etapas, mostrando en cada etapa su duración (segundo código, incompleto).*

```
const ee = require('events').EventEmitter
const emisor=new ee()
function Evento(evento,emitter,n){
  return {
    emit:function(incr){
       n += incr; emisor.emit(evento,emitter,n)
    },
    on:
function(listener){emisor.on(evento,listener)}
  }
}
module.exports=Evento
```

```
const Evento=require("./generadorEventos")
const evento1 = 'e1', evento2 = 'e2'
var incremento=0
const emisor1 = Evento(evento1,'emisor1:  ',0)
const emisor2 = Evento(evento2,'emisor2:  ',")
function visualizar(entidad,evento,dato){
   console.log(entidad,evento+'--> ',dato)
}
emisor1.on(evento1,visualizar)
emisor2.on(evento2,visualizar)
/* ... */ emisiones()
```

a Se soluciona usando la función `setInterval`.

b Para poder mostrar la duración de cada etapa al finalizar la misma, debe pasarse la duración como parámetro al listener del `setTimeout`, o usar una clausura que permita el acceso al valor.

c La variable incremento, que permite llevar el conteo de etapas, se almacena en el evento.

d Para visualizar la duración de cada etapa se utiliza una variable de la función `emisiones`

4 *En la práctica 1 se plantean tres pasos incrementales para construir un proxy programable a partir de uno básico. Si el programador envía el mensaje* `[{remote_ip:44.55.66.77, remote_port:8888}]` *al puerto* `8001` *del proxy programable, indica cuál de las siguientes afirmaciones es cierta:*

a Un navegador web que conecte al puerto `8001` del proxy programable recibirá como respuesta un mensaje `Connection rejected`.

b Un navegador web que conecte al puerto `8001` del proxy programable recibirá como respuesta la que obtedríamos con un navegador que accede a `http://44.55.66.77:8888`

c El componente netClient puede conectar al puerto `8000` del proxy programable y recibir como respuesta la que obtedríamos con un navegador que accede a `http://44.55.66.77:8888`

d Ninguna de las demás opciones es correcta.

5 *A continuación aparece una versión simplificada de uno de los ejemplos Javascript propuestos para su estudio en la primera parte de la práctica 1. Si lo ponemos en ejecución, selecciona qué mostrará en pantalla (cada coma representa un salto de línea):*

```
for (var i=0; i<5; i++)
   (function(){
     setTimeout(function(){
       console.log("A"+i)
     }, i*10)
   })()

for (var k=0; k<3; k++)
   (function(k){
     setTimeout(function(){
       console.log("B"+k)
     }, k*31)
   })(k)
```

a A0, B0, A1, A2, A3, B1, A4, B2

b A0, A1, A2, A3, A4, B0, B1, B2

c A5, B0, A5, A5, A5, B1, A5, B2

d A0, B0, A1, B1, A2, B2, A3, A4

6 *Selecciona la respuesta correcta*

a Cuando dos clientes acceden al proxy, el proxy falla

b Mientras no hay clientes conectando con el proxy, el proxy realiza operaciones `ping` sobre los servidores

c El proxy únicamente conecta con un servidor cuando el cliente envía datos

d Un proxy puede ser cliente de otro proxy