

Examen Ordinario. Teoría. 18 mayo de 2018.

Nombre y Apellidos: _____ DNI: _____

Responde a las siguientes preguntas, de forma breve y concisa. Cada pregunta tiene un valor de **0.5** puntos

1. Tecnologías web: ¿Qué utilizamos para dar estructura a la información? ¿Y estilo? ¿Y para realizar acciones?

Usamos HTML para la estructura, CSS para el estilo y Javascript para las acciones

2. Indica los tipos de conexiones sobre los que HTTP puede funcionar y ordénalos de peor a mejor según su eficiencia

Hay tres tipos de conexión: Conexión no persistente, conexión persistente sin pipelining y conexión persistente con pipelining. La conexión no persistente es la menos eficiente y la conexión no persistente con pipelining la mayor

3. ¿Qué es el DOM?

Es el modelo de objetos del documento. Es una interfaz estándar para acceder a todos los objetos que constituyen la página web, y poder manipularlos

4. ¿Cómo se le asigna un identificador a un objeto del DOM en HTML? Pon un ejemplo

A los objetos del DOM se les asigna una identificación dando un valor al campo id en las etiquetas HTML. Ejemplo: <section id="darth_vader"> Se crea una sección nueva cuya identificación es la cadena darth_vader

5. ¿De qué color sale el mensaje M1 cuando el navegador lo renderiza en la pantalla?

```
<body>
  <p style="color:black">M1</p>
</body>
<head>
  <style>
    p: {color:red}
  </style>
</head>
```

Negro

6. Dado el siguiente fragmento de código, indicar qué se imprime en la consola del navegador

```
function Particula() {  
    this.x = 0;  
    this.y = 0;  
    this.incx = function() {  
        this.x += 3;  
    }  
}  
  
var p1 = new Particula();  
p1.incx()  
p1.incx()  
console.log(p1.x)
```

En la consola se imprimiría el número 6

7. Dado el siguiente código HTML alojado en el fichero index.html de un servidor web, indicar qué peticiones HTTP realiza el cliente cuando se le solicita visualizar esa página

```
<!DOCTYPE html>  
<html>  
<head>  
<link rel="stylesheet" type="text/css" href="mystyle.css"/>  
<script src="main.js"></script>  
</head>  
<body>  
<a href="page2.html"> Go to page 2 </a>  
  
</body>  
</html>
```

En total hace 4 peticiones de tipo GET para bajarse todos los ficheros que necesita del servidor web. La primera es para obtener el fichero index.html. Luego el de estilo, después el de javascript y por último la imagen.

8. En javascript, ¿Se pueden pasar funciones como argumentos de otras funciones?

Si

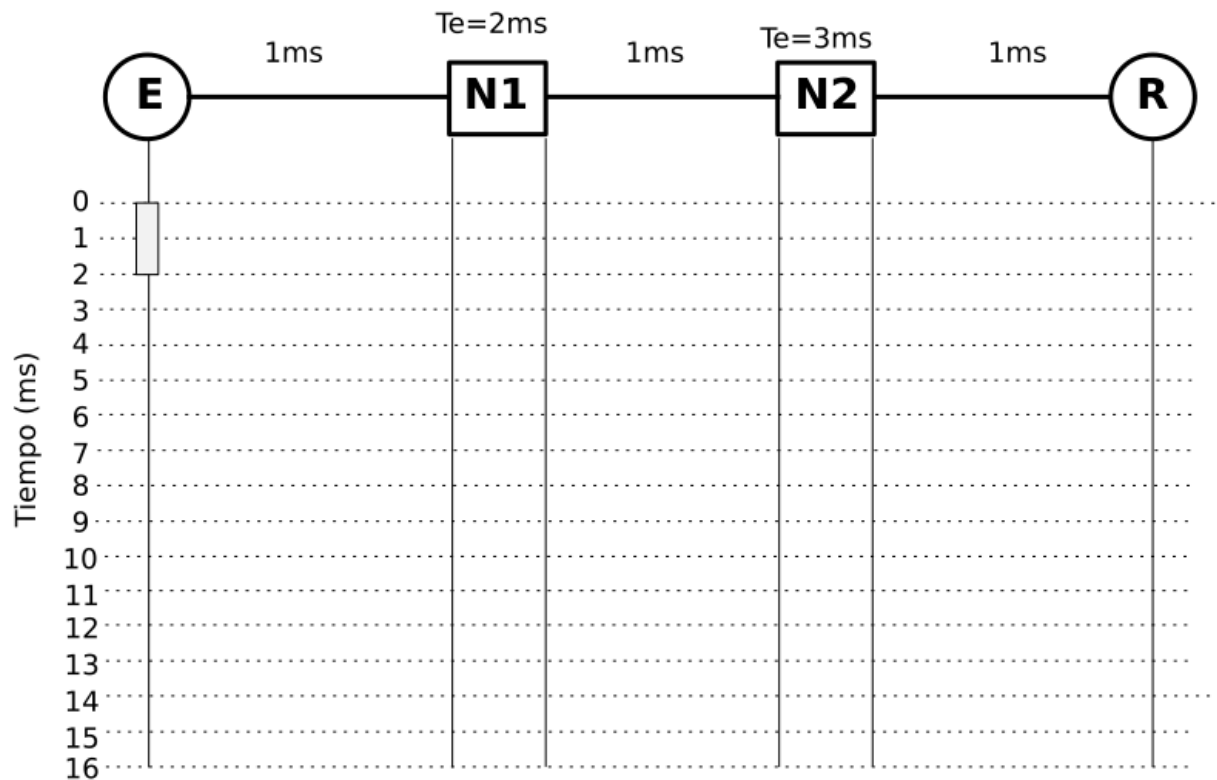
9. Describir el funcionamiento de esta página web

```
<!DOCTYPE html>  
<html>  
<head> <script>
```

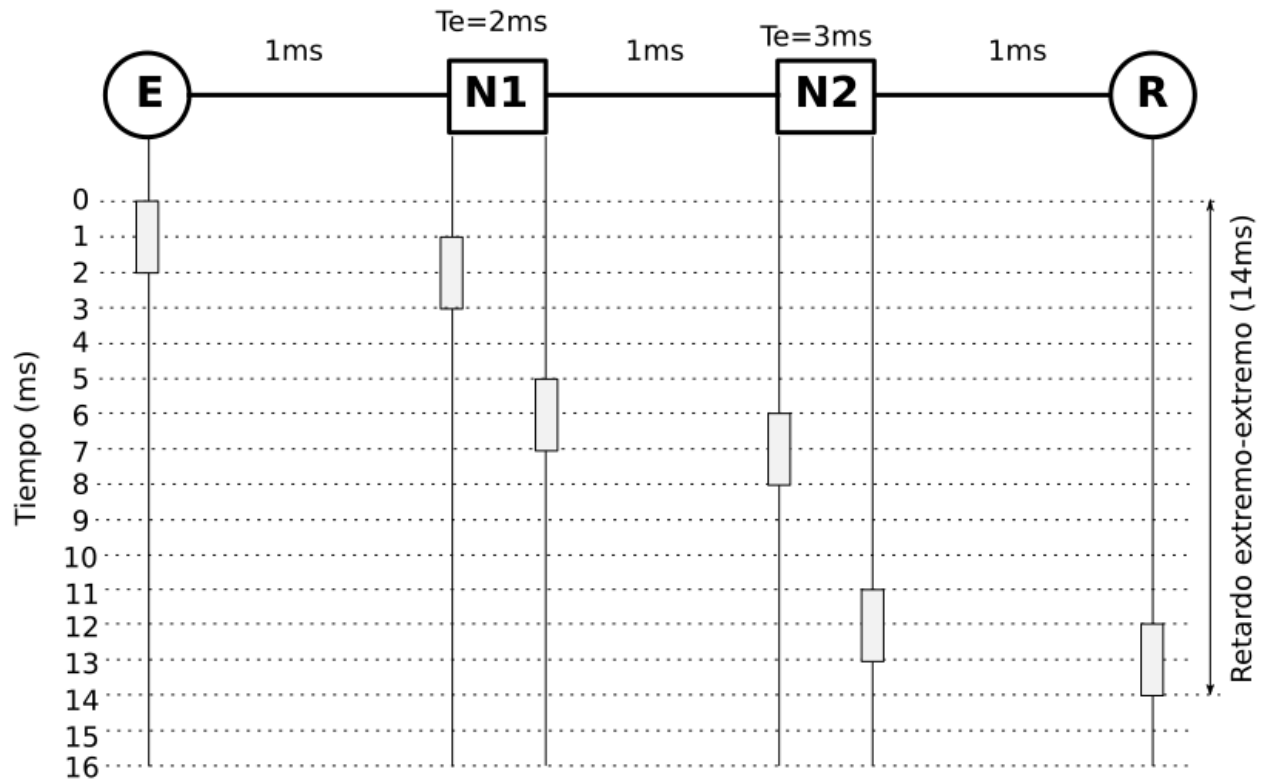
```
function test() {
  setInterval(function() { console.log("Hola") }, 3000);
}
</script> </head>
<body>
  <button onclick="test()">Hola</button>
</body>
</html>
```

La página sólo contiene un botón. Al pulsarlo se invoca la función test() la cual hace que se imprima en la consola del navegador la cadena “Hola” cada 3 segundos

10. En la siguiente red todos los enlaces tienen la misma capacidad, un tiempo de propagación de 1ms, y tiempos de encolamiento de 2ms y 3ms en los nodos 1 y 2 respectivamente. El emisor envía un paquete en el instante 0, que tarda 2ms. Completar el dibujo, colocando el paquete en las líneas temporales de todas las máquinas



Dado que la capacidad de todas las líneas es la misma, el paquete tendrá siempre una longitud de 2ms. Debido al tiempo de propagación, el primer bit tarda 1ms en llegar al nodo 1. Una vez recibido el paquete completo, está en N1 durante 2ms y se empieza a enviar a N2. Tras 1ms se recibe el primer bit. El paquete está en el N2 durante 3ms y se envía finalmente a R



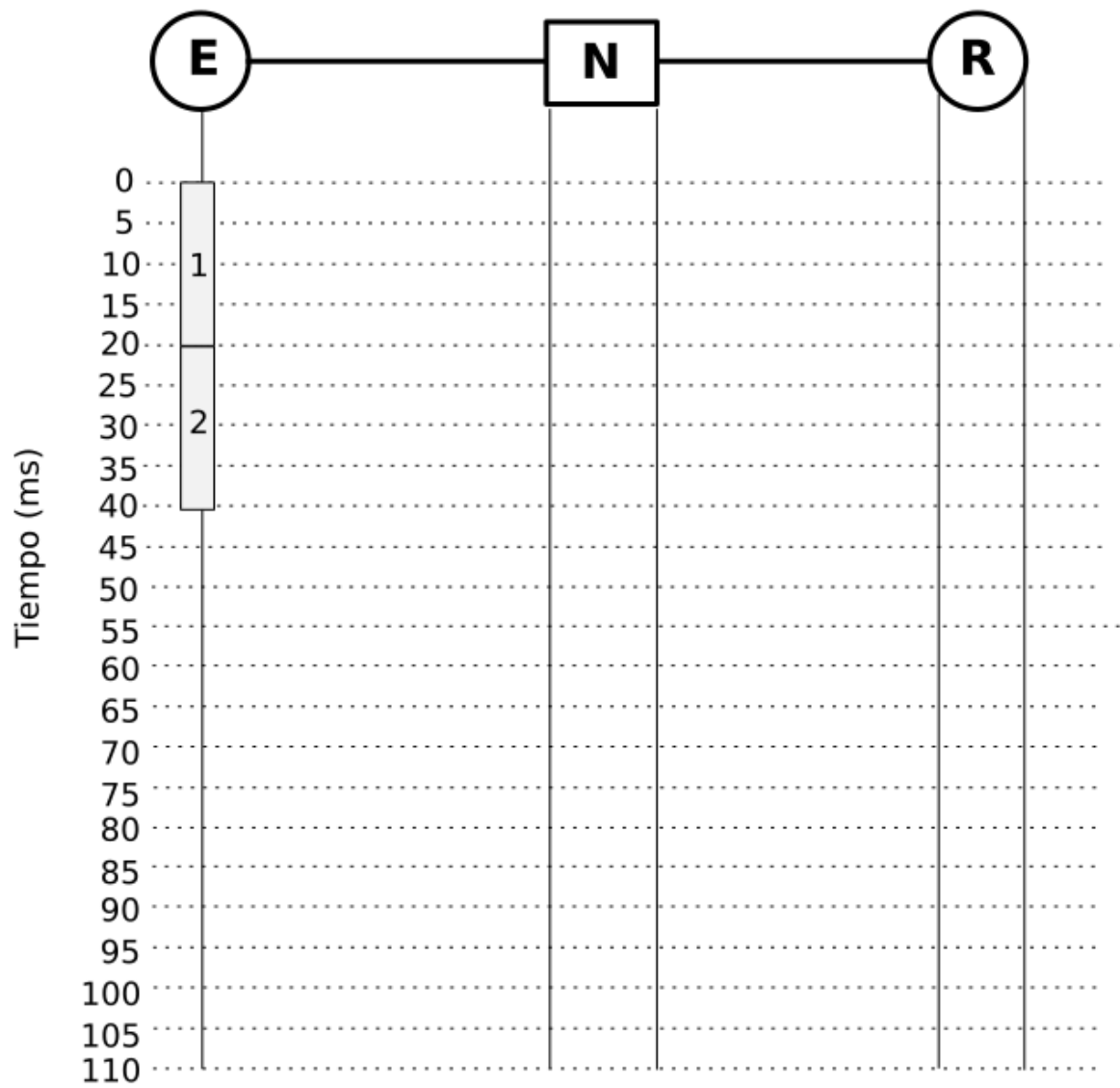
11. ¿Cuánto es el retardo extremo a extremo en la pregunta 10?

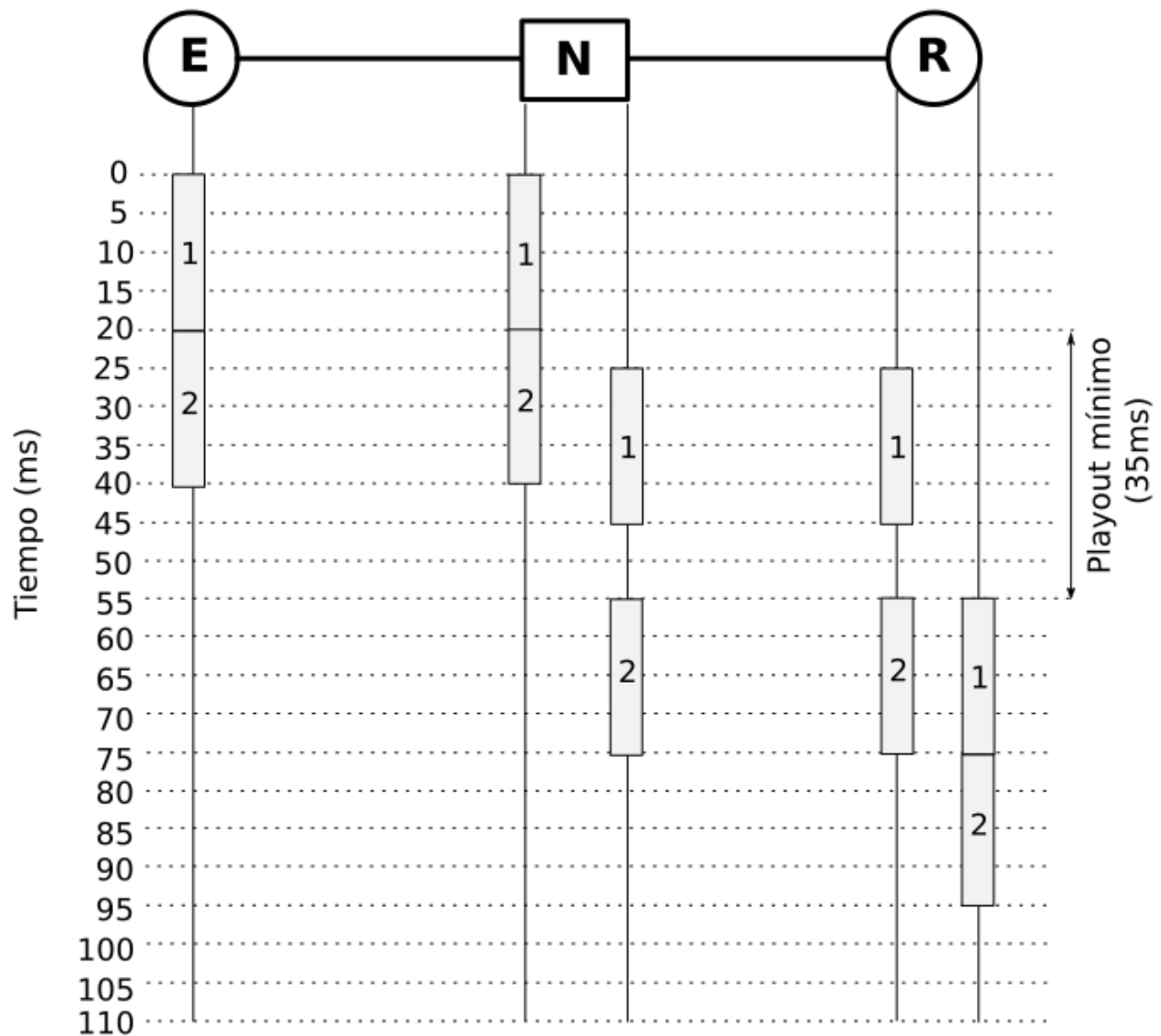
Se ha mostrado en la figura anterior. El retardo extremo a extremo es de 14ms

12. ¿Qué es el tiempo de *playout* ?

Es el tiempo transcurrido desde que se envía un paquete en el emisor y se empieza a reproducir en el receptor

13. Se tiene la siguiente red, donde hay un emisor, un nodo intermedio y un receptor. El tiempo de propagación lo supondremos 0. En emisor y receptor se está ejecutando una **aplicación interactiva**, que en un momento determinado envía paquetes multimedia de 20ms. El tiempo de encolamiento en el nodo del primer paquete es de 5ms y de 15ms el segundo. Calcular el tiempo de *playout* **mínimo** para reproducir correctamente los dos primeros paquetes





Como el tiempo de propagación se desprecia, el paquete 1 se empieza a recibir en el instante 0 en el nodo intermedio N. Tras 5ms en la cola, se empieza a enviar a R, que lo recibe. El paquete 2 se recibe en N a continuación del 1, pero ahora está en la cola durante 15ms, y se envía a R. El emisor ha enviado los paquetes 1 y 2 seguidos, pero en el receptor hay una interrupción de 10ms. Para que la reproducción sea continua, hay que retener en el buffer el paquete 1 hasta el instante 55ms. Si se empieza a reproducir a partir de ahí, entonces ya no hay saltos. El tiempo de playout deberá ser de al menos 35ms. El playout mínimo son, por tanto, 35ms

14. En el caso anterior. ¿Cuál sería el playout **máximo**?

Como se trata de una aplicación interactiva, el tiempo de playout no puede ser mayor de **450ms**, para que no se degrade la calidad

15. En el caso anterior, si la aplicación fuese de **streaming** en vez de interactiva, ¿Cuál sería el *playout máximo*?

Las aplicaciones de streaming son menos restrictivas que las interactivas, por lo que el tiempo de playout máximo puede ser de **10 segundos**

16. Comenta sobre la viabilidad de utilizar técnicas de recuperación de errores basadas en retransmisión para las aplicaciones de datos, multimedias interactivas y multimedias de streaming

La forma clásica de recuperar tramas corruptas o no recibidas es mediante retransmisión. Esto es lo que hace TCP, que nos garantiza que la recepción de datos correctos y ordenados. Es viable su uso con las aplicaciones de datos. En el caso de las aplicaciones multimedia, se puede utilizar en las de streaming, ya que son menos restrictivas. Sin embargo no es factible para las interactivas

17. Indica las tres técnicas de recuperación sin retransmisión principales en aplicaciones multimedia

Las técnicas de recuperación de errores sin retransmisión principales son la FEC, el entrelazado y la reconstrucción

18. Un emisor y receptor multimedia están usando la técnica FEC genérica para corregir errores. Usan paquetes de 5 bits y grupos de 6 paquetes. Se produce un problema en la transmisión, por lo que el receptor sólo recibe 5 de los 6 paquetes del grupo, cuyos valores son: 00100, 11111, 10101, 00000, 01110. Reconstruye el paquete perdido

El criterio que se usa es de paridad par. Tiene que haber siempre un número par de unos en los bits con el mismo peso. Los colocamos en filas y aplicando el criterio de paridad par deducimos el valor del paquete perdido

00100
11111
10101
00000
01110
00000

Aplicando este criterio, podemos calcular los bits del paquete que falta: 00000

19. Indica las ventajas y desventajas de utilizar la técnica del entrelazado para recuperación de errores

La principal ventaja es que no aumenta el caudal utilizado, ya que no se añade redundancia. La desventaja es que requiere de un retardo adicional porque hay que meter en el buffer los paquetes recibidos para deshacer el entrelazado antes de su reproducción

20. Indica todas las posibilidades que conozcas a nivel de arquitectura para el diseño de una aplicación de streaming en directo

Se podría construir la aplicación directamente sobre UDP, también TCP, sobre RTP o incluso HTTP. En cada caso la aplicación tendría que implementar la funcionalidad que faltase, pero sería viable