



Ingeniería Técnica
Informática de Sistemas
Departamento de Informática

Ingeniería
Telemática

REDES – Primer parcial		Modelo A
APELLIDOS:		20 – Junio - 2007
NOMBRE:		DNI:

CUESTIÓN 1.- Define los siguientes términos: [0.5 puntos]

Protocolo	
Distorsión de retardo	
MFSK	
Scrambling	
ADSL	

CUESTIÓN 2.- Indica las capas de TCP/IP, las funciones y dos protocolos de cada una. [0.5 puntos]

Capas	Funciones	Protocolos

CUESTIÓN 3.- Indica la expresión general de una onda sinusoidal y define sus parámetros. [0.5 puntos]

Expresión general:	

CUESTIÓN 4.- ¿A qué se denomina velocidad de modulación?. Indica su fórmula junto con el significado de los parámetros. [0.5 puntos]

CUESTIÓN 5.- ¿En qué consiste la técnica modulación por impulsos codificados?. [0.75 puntos]

PROBLEMA 1.- Codifíquese, utilizando **B8ZS**, la transmisión de la siguiente cadena binaria, teniendo en cuenta que se empezará con un pulso positivo. En la segunda fila complete la polaridad del pulso representado en las filas 4 y 5 utilizando la siguiente notación: [0.5 puntos]

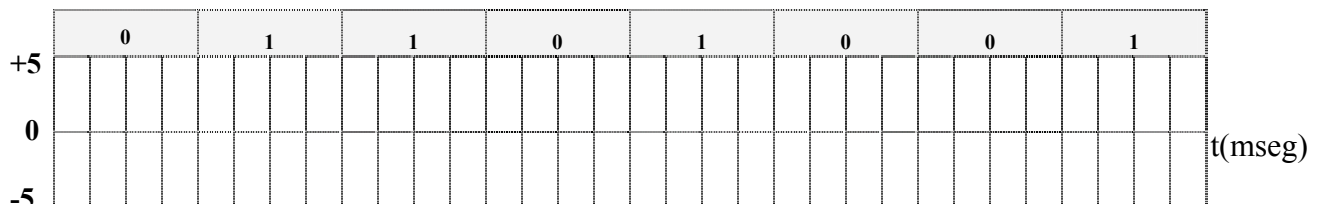
- B** ⇒ Impulso bipolar válido
- V** ⇒ Impulso de violación de paridad
- 0** ⇒ Ausencia de señal

1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

PROBLEMA 2.- Se quiere transmitir en **ASK** la información digital que se indica a un régimen binario de **R = 2Kbps**. Para ello se utiliza un módem a una frecuencia de portadora de **f_C = 3KHz** con la siguiente asignación de fases:

- 1** ⇒ A = 5 voltios, fase = π
- 0** ⇒ A = 0 voltios

Represente en la figura la señal de salida resultante, indicando en la base de tiempos la duración de cada bit. **Justifíquese la respuesta.** [0.5 puntos]



PROBLEMA 3.- Se desea realizar una TDM síncrona sobre 6 fuentes analógicas de 4KHz cada una y 6 fuentes digitales de 8,5Kbps cada una. En el caso de las fuentes digitales se desea unificar su velocidad a 9Kbps. Para ello se cuenta con un conversor A/D de 6 bits. Realizar el esquema general y explicar su funcionamiento. [0.75 puntos]

PRÁCTICA [0.5 puntos]

Se desea realizar un programa cliente-servidor de acuerdo al siguiente protocolo:

1. El servidor una vez recibida una solicitud de conexión de un cliente, le solicite el nombre de usuario, para ello enviará al cliente la cadena “login”, que deberá ser visualizada en el cliente.
2. El cliente introducirá por teclado su login, el cual será enviado al servidor.
3. El servidor tras recibirlo, le solicitará una password, a través del envío de la cadena “password”, que deberá ser visualizada en el cliente.
4. El cliente introducirá por teclado su contraseña y se la enviará al servidor.
5. Una vez que el servidor la ha recibido comprobará si es o no correcta. Si lo es enviará al cliente la cadena “Bienvenido” y en caso contrario la cadena “Adios”.
6. En ambos casos cerrar la conexión en ambos extremos.

APLICACIÓN SERVIDOR

```
import java.net.*;
import java.io.*;

public class ServidorParlante{
    public static final int PUERTO = 
    public static void main(String [] args) throws IOException{
        //crear el socket del servidor
```

```

[ ]
try{
    [ ] //Quedar a la espera de alguna conexión
    try{
        BufferedReader entrada=
            new BufferedReader(new InputStreamReader(socket.getInputStream()));

        PrintWriter salida= new PrintWriter(new BufferedWriter
            (new OutputStreamWriter(socket.getOutputStream())),true);
        while (true){
            //Escribir el protocolo

            [ ]

        }
    }finally{
        //Cerrar connexion con el cliente
        [ ]
    }
}finally{
    //Cerrar socket del servidor
    [ ]
}
}}

```

APLICACIÓN CLIENTE

```

import java.io.*;
import java.net.*;

public class ClienteParlante{

    public static final int PUERTO = [ ];
    public static void main(String [] args) throws IOException{

        InetAddress addr=InetAddress.getByName(null);
        //Crear el socket del cliente

        [ ]
    }
}

```

```
try{
    BufferedReader entrada= new BufferedReader
        (new InputStreamReader(socket.getInputStream()));
    //La salida vaciada automáticamente por PrintWriter:
    PrintWriter salida= new PrintWriter(new BufferedWriter
        (new OutputStreamWriter(socket.getOutputStream()),true);
    //variable para leer del teclado
    BufferedReader in = new BufferedReader (new InputStreamReader(System.in));

    //Escribir el protocolo

}
}finally{
    //Cerrar el socket
}
}
```