

## BOLETÍN 2 SOCKETS

1. Crea las siguientes clases con los siguientes atributos, los constructores y los métodos get y set necesarios:

Clase Asignatura:

```
int id;  
String nombreAsig;
```

Clase Especialidad:

```
int id;  
String nombreEsp;
```

Clase Profesor:

```
int idProfesor;  
String nombre;  
Asignatura [] asignaturas;  
Especialidad esp;
```

Un profesor podrá impartir hasta 3 asignaturas.

Utilizando sockets TCP, implementar un programa servidor que inicialice un array de 5 objetos de tipo Profesor. El servidor aceptará conexiones de clientes en un bucle infinito. Cada vez que se conecte un cliente, el servidor le asignará un id, que empezará en 1 e irá incrementándose cada vez que se conecta un nuevo cliente. El servidor recibirá del cliente un idProfesor y le devolverá el objeto Profesor asociado.

Crea un programa cliente en el que se introduzca por teclado el *idProfesor* que se desea consultar. Dicho programa recogerá datos hasta recibir un \* por parte del usuario.

Si el *idProfesor* no se encuentra registrado, el servidor le devolverá un objeto *Profesor* con datos vacíos

2. Crea un programa cliente usando sockets UDP que envíe el texto escrito desde la entrada estándar al servidor. El servidor le devolverá la cadena en mayúsculas. El proceso de entrada de datos finalizará cuando el cliente introduzca un asterisco.

Crea un programa servidor que reciba cadenas de caracteres, las muestre en pantalla y se las envíe al emisor en mayúscula. El proceso servidor finalizará cuando reciba un asterisco.

Establece un tiempo de espera de 5000ms con el método `setSoTimeout` para hacer que el método `receive()` del programa cliente se bloquee. Pasado ese tiempo, controlar si no se reciben datos lanzando la excepción `InterruptedException`, en cuyo caso visualiza un mensaje indicando que el paquete se ha perdido.

3. Crea una clase Java llamada *Numeros* que defina 3 atributos , uno de ellos entero, y los otros 2 de tipo *long*:

```
int numero;  
long cuadrado;  
long cubo;
```

Define un constructor con parámetros y otro sin parámetros. Define los métodos *get* y *set* de los atributos. Crea un programa cliente que introduzca por teclado un número e inicialice un objeto *Numeros*, el atributo *numero* debe contener el número introducido por teclado. Debe enviar ese objeto al programa servidor. El proceso se repetirá mientras el número introducido por teclado sea mayor que 0.

Crea un programa servidor que reciba un objeto *Numeros*. Debe calcular el cuadrado y el cubo del atributo *numero* y debe enviar el objeto al cliente con los cálculos realizados, el cuadrado y el cubo en sus atributos respectivos. El cliente recibirá el objeto y visualizará el cuadrado y el cubo del número introducido por teclado. El programa servidor finalizará cuando el número recibido en el objeto *Numeros* sea menor o igual que 0.

Controlar posibles errores, por ejemplo si ejecutamos el cliente y el servidor no está iniciado, o si estando el servidor ejecutándose ocurre algún error en el cliente, o este finaliza inesperadamente, etc.

4. Usando sockets UDP, realiza un programa servidor que espere un datagrama de un cliente. El cliente le envía un objeto *Persona* que previamente había inicializado. El servidor modifica los datos del objeto *Persona* y se lo envía de vuelta al cliente. Visualiza los datos del objeto *Persona* tanto en el programa cliente cuando los envía y los recibe como en el programa servidor cuando los recibe y los envía modificados.

5. Crea una clase de nombre *Curso*, con los siguientes atributos:

```
String id;  
String descripcion;
```

Crea otra clase de nombre *Alumno*, con los siguientes atributos:

```
String idalumno;  
String nombre;  
Curso curso;  
int nota;
```

Crea en las clases anteriores los constructores y métodos *get* y *set* necesarios.

Utilizando sockets UDP crea un programa servidor que inicialice un array de 5 objetos de tipo *Alumno*.

El servidor, en un bucle infinito, solicitará al cliente un *idAlumno* y le devolverá el objeto *Alumno* que corresponda.

Crea un programa cliente en el que se introduzca por teclado el *idAlumno* que se desea consultar. Dicho programa recogerá datos hasta recibir un \* por parte del usuario.

Si el *idAlumno* no se encuentra registrado, el servidor le devolverá un objeto *Alumno* con datos vacíos.

6. Realiza un programa servidor que escuche en el puerto 12345. Cada vez que se conecte un cliente se creará un nuevo hilo para atenderlo. Se mostrará en la consola del servidor la dirección IP y el puerto remoto del cliente que se conecta. Se deberá notificar también cuando un cliente se desconecte.

En el hilo que atiende al cliente, se reciben cadenas de caracteres que, mientras sean distintas de "\*", se devolverán al cliente transformadas a mayúsculas.