

PSP - Unidad 3 - Examen

Índice de contenido:

1. Cliente - Servidor.....	0
2. Listar ficheros.....	1
3. Criterios de evaluación (CE).....	1

1. Cliente - Servidor

Crea un programa usando sockets que permita a un cliente enviar mensajes de texto a un servidor.

Para ello debes crear una clase cliente y otra servidor. Obviamente, esta aplicación funcionará en local.

- **(6 puntos)** La clase **servidor** esperará a que se conecte un cliente y recibirá los mensajes de texto que éste le envíe.

Por cada mensaje recibido, el servidor enviará un mensaje al cliente indicando que lo ha recibido correctamente.

Al arrancar y antes de esperar la conexión de los clientes, el servidor pedirá que se especifique un **directorio base** y deberá permitir al usuario indicarlo **fácilmente**.

Además de los textos, el servidor puede recibir algunas órdenes o comandos que se identifican porque empiezan por almohadilla (#).

Las órdenes que el servidor puede reconocer se pueden incluir en un enum o como constantes de la clase servidor y serán las siguientes:

#fin - para indicar al servidor que finaliza la conexión.

#info - para que el servidor muestre la información de conexión: host y puerto.

#list - el cliente pide al servidor que le responda enviando la lista de los ficheros¹ del directorio del servidor.

- **(4 puntos)** La clase **cliente** podrá enviar mensajes al servidor, que pueden ser textos u órdenes como las indicadas arriba. de forma que el servidor mostrará los mensajes enviados por el usuario y, además, responderá al cliente para confirmar que los ha recibido.

¹ Se da en este enunciado un pequeño recordatorio sobre esto.

2. Listar ficheros.

La clase `File` sirve para crear un objeto que represente a cualquier objeto (fichero o directorio).

```
File directorio = new File("ruta a un directorio");
```

Para que todo sea correcto, hay que comprobar si ese objeto `directorio` es realmente un directorio (`isDirectory()`).

Después se puede obtener la lista de ficheros con el método `directorio.listFiles()` y mediante un bucle mostrar la lista de nombres de esos ficheros.

Esto debe meterse en una cadena para que el servidor se lo envíe al cliente.

3. Criterios de evaluación (CE)

Este ejercicio permite evaluar el **RA 3. Programa mecanismos de comunicación en red empleando sockets y analizando el escenario de ejecución** con los siguientes criterios:

3.e) Se han utilizado sockets para programar una aplicación cliente que se comunique con un servidor.	14%
3.f) Se ha desarrollado una aplicación servidor en red y verificado su funcionamiento.	14%