

DSI 2010: Práctica 2

1. Descripción

La facultad de informática, en un intento para mitigar el uso excesivo de sistemas de mensajería instantánea en los laboratorios de prácticas, encarga el diseño de un sistema interno llamado FicTalk. El sistema está formado por una serie de clientes que se conectan a un servidor central, el cual recibe todos los **mensajes** y se encarga de dirigirlos al cliente que corresponda.

Cada cliente guarda una lista de **contactos**. Cada contacto posee un identificador, que permite distinguir a los distintos usuarios del sistema, y un nombre, que solo se utiliza en ese cliente en concreto como nombre que se muestra a través de la interfaz. Los clientes pueden enviar y recibir **mensajes** a través del servidor. Un mensaje tiene un texto con el contenido del mensaje, un identificador del origen del mensaje, y uno o más destinatarios.

El servidor se encarga de reenviar los **mensajes** a sus destinatarios.

Se desea añadir una serie de mejoras:

- La permanente falta de financiación de la universidad ha obligado a buscar fuentes alternativas de ingresos. Para ello pide añadir al sistema la posibilidad de añadir **publicidad** a los mensajes que los usuarios envían. Esta publicidad deberá ser añadida por el servidor sin modificar los mensajes originales.
- Para facilitar la comunicación entre grupos de prácticas, se quiere añadir la posibilidad de definir **grupos** de contactos. Un grupo estará formado por una serie de contactos individuales u otros grupos, pero un mismo contacto no podrá estar en más de un grupo. Cuando se abre una conversación con un grupo de usuarios, la comunicación llegará a todos sus miembros.
- Para evitar el uso excesivo del sistema por parte de los usuarios, se pide añadir al sistema un **limitador**, que impida enviar a cada usuario más de una cierta cantidad de mensajes por minuto.
- Implementar un **que** impida a determinados usuarios comunicarse entre ellos, ignorando los mensajes que estos se envíen entre sí.
- La natural curiosidad de los operadores del sistema por las conversaciones ajenas les lleva a solicitar la implementación de un sistema de **log** para el servidor que almacene todos los mensajes en un fichero de log. En el fichero de log deberá aparecer toda la información relevante de cada mensaje.

Se pide:

- Elaborar el diagrama de clases del sistema descrito en formatos MagicDraw y PNG.
- Implementar en el lenguaje java las mejoras propuestas.

Para facilitar la realización de la práctica se proporciona una interfaz para los clientes, junto con la funcionalidad básica para los clientes y servidores. Como ejemplo de funcionamiento se proporcionan implementaciones para arrancar un servidor y dos clientes y un fichero `build.xml` para compilar y ejecutar la práctica mediante ant:

- `ant compile` compila todos los ficheros de la práctica.
- `ant run-server -Dport=2000` arranca un servidor en la máquina local escuchando en el puerto 2000.
- `ant run-client1 -DIP=127.0.0.1 -Dport=2000` conecta a un cliente con el servidor situado en la máquina 127.0.0.1 y escuchando en el puerto `port`. El cliente 2 se arranca de una forma similar.

Se pueden añadir nuevos clientes utilizando como base los proporcionados.

2. Entrega

Deberá hacerse al menos un commit sobre el repositorio para cada una de las partes de la práctica.

Debido a problemas con clases perdidas con los grupos del viernes, la fecha de entrega se comunicará a través del campus virtual, pero en ningún caso será antes de semana santa.

3. Enlaces al padre en el patrón Composición

En su forma habitual el patrón composición no permite comprobar de forma sencilla si un elemento simple está en más de un elemento compuesto. En algunos casos esto puede ser necesario, y la forma más habitual de hacerlo es utilizando un atributo que enlaza al padre, y métodos para consultarlo y establecerlo. La visibilidad se limita tanto como es posible para el método establecer (normalmente se pone como protegido).

La operación de agregar elementos puede utilizar estos métodos para comprobar si el elemento ya estaba en otra colección y realizar alguna acción (bien eliminarlo de la anterior antes de añadir utilizando el método para establecer el padre, bien lanzar un error).