

Sustitución del callback RMI por uno basado en JMS

El desarrollo de nuestro Twitter lo hicimos intentando modularizar y factorizar al máximo el código, para que así cualquier cambio fuese sencillo de realizar.

Para el manejo del callback el cliente tiene una clase intermedia que hacía de intermediario entre los eventos que llegaban del servidor y la interfaz. Esta clase se llama *ClientCallbackListener* tanto para la conexión con RMI con nuestro servidor o la conexión mediante Jtwitter ya que ha sido desarrollada para ser utilizada por ambos implementando la interfaz *Astream.IListen* y es la que se pasa a nuestro servidor como callback o se añade como listener del stream de JTwitter.

El equivalente a esta clase para JMS se llama *ClientTopicListener*, implementa *MessageListener* y se encarga en su constructor de conectar con el topic “Eventos” pasándole un string selector al suscribirse al topic para recibir solo los mensajes que concuerden con ese selector. El selector filtrará en base al atributo *userId* que se le añade a todos los mensajes y que consiste en el id del usuario con el que está relacionado el mensaje. Así, los clientes escucharán tanto los mensajes relacionados con su propio usuario como los de los usuarios a los que sigue y los mensajes considerados como globales (con *userId* = -1).

De esta manera, con solo cambiar una clase hemos conseguido que el cliente pase a funcionar escuchando los eventos mediante JMS.

En el lado del servidor los mensajes se enviaban por el callback mediante una función que se encuentra en *TwitterInitImpl*:

```
public void sendThroughCallback(Serializable event, Long id_dest)
```

Así, para realizar el cambio a JMS solo hubo que sustituir esta función por una que utilizase el topic que llamamos *sendThrougTopic* y cambiar las partes del código que utilizaban la función anterior por la nueva.