

Middleware

Práctica 1: Aplicaciones basadas en Componentes Distribuidos

Curso 2014-2015

Introducción

En esta primera práctica de la asignatura se va a desarrollar una aplicación distribuida que implemente una red social basada en microblogging, utilizando tecnologías Java (SE y EE, **versión 7** o superior).

Descripción de la práctica

La red social que se desea diseñar e implementar debe funcionar de forma similar a **Twitter** [1], si bien únicamente se tendrá que desarrollar una pequeña parte de su funcionalidad, de interés para la asignatura.

Para hacer la práctica lo más realista posible (e interesante), se recomienda seguir la documentación ofrecida por Twitter en su página web de desarrolladores [2], y en concreto en relación con el modelo de datos (mirar los *API Objects* [3]), y en concreto con los objetos (clases) *Tweet* (o *Status*) y *User*.

La funcionalidad pedida para la aplicación es la siguiente:

- La aplicación debe permitir que un usuario se registre, se conecte y se desconecte.
- Un usuario debe poder establecer su estado, utilizando texto y no más de 140 caracteres. Se debe controlar el tamaño del mensaje, y en caso de superar los 140 caracteres, truncarlo y mostrar unos puntos suspensivos.
- Los usuarios pueden seguir a otros usuarios, de modo que cuando el usuario al que siguen actualiza su estado, éste se les notifica al instante apareciendo en el primer lugar de su *Timeline*.
- Los usuarios tienen un *Timeline* o colección de sus *Status* más recientes y los de los usuarios a los que sigue.
- Los usuarios deben poder establecer su perfil de usuario, indicando su nombre real, nombre de usuario, foto, dirección web, etc...
- Los usuarios deben poder hacer búsquedas para localizar otros usuarios y poder seguirlos.
- Un usuario puede mandar mensajes directos (privados) a otro usuario.

Normas básicas

En base a todo lo anterior, se pide desarrollar una aplicación distribuida que implemente la funcionalidad básica de la red social comentada, respetando las siguientes normas:

- **Uso de Java RMI**

La aplicación distribuida debe estar formada por un servidor cuyos servicios se deben ofrecer al cliente a través de una o varias APIs remotas utilizando Java RMI. Esto incluye, además de la funcionalidad indicada anteriormente, la notificación de las actualizaciones de estado a los usuarios utilizando la técnica de *callback*.

Téngase en cuenta que es tarea del alumno decidir qué clases deben ofrecer qué métodos como métodos remotos y declarar e implementar las interfaces remotas correspondientes.

Además:

- Se podrán crear todas las clases, métodos y atributos que el alumno considere necesarios, así como añadir funcionalidad extra a semejanza del Twitter real.
- No es obligatorio, pero se valorará además la creación de la aplicación cliente como una aplicación gráfica de escritorio (utilizando *Swing* o *JavaFX* por ejemplo), o mejor aún, creando un cliente web mediante Servlets y JSPs.
- Respecto a la persistencia de los datos en el servidor, también se puede implementar como se desee, es decir, se puede utilizar una base de datos, Hibernate, JPA, o simplemente objetos en memoria.
- El desarrollo de este trabajo es obligatorio para aprobar la asignatura, y servirá de base para la realización del resto de prácticas de la asignatura.

Grupos de prácticas

El trabajo se desarrollará en grupos de 2 o 3 personas, aunque también se aceptarán trabajos individuales de forma justificada.

Los grupos de 3 personas tendrán que realizar, además, el siguiente trabajo extra:

- Estudiar una API Java para Twitter [4].
- Analizar si es posible integrar el cliente Twitter desarrollado en la red Twitter real [1], indicando las funcionalidades permitidas y los problemas encontrados.
- Una vez concluido el estudio y análisis pedido, los alumnos tendrán que adaptar el código del cliente desarrollado en la práctica para conectarse a la red social Twitter real [1].

Entrega del trabajo:

- La fecha establecida para entregar la práctica es el **9 de noviembre de 2014 a las 23:55**.
- El código fuente de la aplicación, así como una breve explicación de cómo ejecutar el sistema distribuido desarrollado, se subirá a través de una tarea que se publicará en Moodle antes de la fecha de presentación anteriormente comentada.
- Se entregará una memoria (breve) en formato pdf que contendrá al menos los siguientes apartados:
 - Definición de la interfaz del sistema.
 - Explicación de la API RMI diseñada
 - Diagrama de clases de los componentes del sistema distribuido.
 - Estudio y análisis de la API Java para Twitter (solo grupos de 3 personas)
 - Opinión personal

Esta memoria se entregará conjuntamente con el código fuente.

Para dudas o consultas sobre el trabajo, **se deberá utilizar el foro disponible en el aula virtual** para tal efecto. Se valorarán las contribuciones del alumno en dicho foro, no solo preguntando, sino también respondiendo a las dudas de los compañeros. Por supuesto, también es posible ponerse en contacto con el profesor Rafael Fernández en el despacho 4310 o en la dirección de correo electrónico rfernandez@fi.upm.es.

Referencias

- [1] Twitter: <http://www.twitter.com>
- [2] Twitter Documentation: <https://dev.twitter.com/overview/documentation>
- [3] API Overview: <https://dev.twitter.com/overview/api>
- [4] API Java Twitter: <http://www.winterwell.com/software/jtwitter/javadoc/>