Universidad Nacional de Educación a Distancia
E.T.S.I. Informática
Dpto. de Sistemas de Comunicación y Control

PRÁCTICA
DE
Sistemas Distribuidos

(Grado en Ingeniería Informática)

Sistema básico de microblogging tipo Twitter® usando Java RMI

Curso 2018 - 2019

INTRODUCCIÓN

Este documento contiene el enunciado de la práctica obligatoria de laboratorio correspondiente a la asignatura de Sistemas Distribuidos del Grado de Ingeniería Informática. Lea detenidamente toda la memoria varias veces con el fin de que sus dudas queden resueltas. Si aún así sigue teniendo alguna duda, por favor consúltela a través del foro de la asignatura en el curso virtual.

El software mínimo necesario para la realización de la práctica es el siguiente:

- Kit de desarrollo de Java 11 JDK. Disponible en el servidor de la empresa Oracle, (http://www.oracle.com/technetwork/java/javase/downloads/index.html) actual propietaria de la tecnología. En esa dirección pueden encontrar además de la documentación oficial de la API del lenguaje, valiosa información como tutoriales, artículos técnicos, etc.
- Entorno de desarrollo IDE. Aunque no es necesario (se pueden editar los ficheros fuentes con un sencillo editor como el notepad y compilar/ejecutar desde la línea de comandos) pero sí se recomienda su uso para acelerar el desarrollo. Además, los editores poseen potentes depuradores que prestan un excelente servicio a la hora de depurar las aplicaciones. Se puede recurrir a cualquier entorno siempre que éste genere código 100% Java. El equipo docente recomienda:

  - Eclipse (http://www.eclipse.org/downloads/)

Les recordamos que la entrega de la práctica es **obligatoria** para todos los estudiantes. Además, es imprescindible aprobar la práctica para aprobar la asignatura. Le recomendamos la lectura detenida de la guía de curso para aclarar cualquier duda sobre los criterios de evaluación.

Una vez entregada la práctica con todo el material requerido, el equipo docente someterá a los programas que se entreguen a un test para comprobar su correcto funcionamiento. Por tanto, se recomienda probar todo el software antes de enviarlo.

Si en la convocatoria de febrero supera la práctica obligatoria pero no supera el examen presencial, se le conservará la calificación de la práctica para la convocatoria extraordinaria de septiembre del curso actual pero **nunca** para cursos posteriores. Análogamente sucederá con la nota del examen presencial.

No se conservan calificaciones de ninguna prueba, ya sea teórica o práctica, para cursos posteriores.

Las prácticas se realizan **individualmente**, no se aceptan grupos de trabajo. La detección de una práctica copiada de cualquier fuente incluida Internet, obligará al equipo docente a ponerlo en conocimiento del Servicio de Inspección de la UNED para

que proceda a la apertura de expediente académico. Evidentemente, pueden consultar cualquier tipo de dudas, cuestiones, mañas, etc. a través de los foros en el curso virtual de la asignatura.

Lea detalladamente el resto de la memoria ya que se le indica claramente la forma de realizar la práctica, el material que hay que entregar, y los plazos para hacerlo. Para cualquier aclaración, no dude en ponerse en contacto con los miembros del equipo docente de la asignatura utilizando los medios que se indican en la Guía del curso.

ENUNCIADO

El propósito de la práctica es el desarrollo de un software que implemente un sistema básico de microblogging al estilo del famoso Twitter® usando Java RMI.

En este sistema actuaran tres tipos de actores cuyas funciones se listan a continuación:

1.- Servidor: La entidad Servidor se encarga de controlar el proceso de autenticación de los usuarios del sistema y gestión de sus mensajes (los cuales vamos a llamar **trinos**), para ello hace uso de dos servicios:

- Servicio Autenticación: Se encarga de registrar y de autenticar a los usuarios del sistema. La operación de registro consiste en que los usuarios introducen en el sistema por primera vez sus datos personales (nombre, nick (apodo) y password), siendo el nick el identificador único de usuario. Es decir, no pueden existir dos usuarios en el sistema con el mismo nick. Una vez registrado, el usuario puede acceder al sistema haciendo login utilizando este servicio de Autenticación.
- Servicio Gestor: Este servicio se encarga de gestionar todas las operaciones de los usuarios en relación a enviar trinos y hacerse seguidor de otros usuarios. Cuando un usuario se hace seguidor de otro usuario, el primero recibe automáticamente los trinos del segundo cuando los publica.

2.- Base de datos: Esta entidad es la encargada de almacenar todos los datos del sistema: Usuarios, Seguidores, Trinos, ...; Sólo la entidad Servidor puede consumir el servicio que suministra esta entidad y cuya funcionalidad se describe a continuación.

- Servicio Datos: Este servicio hará las funciones de una base de datos que relacione Usuarios-Seguidores-Trinos. Es decir, mantendrá la lista de usuarios registrados y/o conectados al sistema, junto con sus seguidores y trinos; y los relacionará permitiendo operaciones típicas de consulta, añadir y borrado. Los dos servicios anteriores (Servicio Autenticación y Servicio Gestor) harán uso de este servicio para realizar las operaciones sobre el estado de los usuarios del sistema y sus seguidores. Aunque podría usarse un sistema de gestión de bases de datos (SGBD) para implementar este servicio, esto haría muy complejo el desarrollo de la práctica y no atendería a los objetivos de la asignatura. Así pues, **el equipo docente recomienda** para la implementación del servicio las clases *List* y *HashMap* de Java.

3.- Usuario: Son los actores principales del sistema e interactúan entre ellos enviándose trinos y haciéndose seguidores unos de otros. Los usuarios se registran en el sistema a través de la entidad Servidor. Una ver registrados se logean usando también esta entidad Servidor para poder enviar trinos y hacerse seguidores de otros usuarios. La entidad Usuario debe implementar un servicio para recibir los trinos de aquellos usuarios a los que siguen:

- Servicio CallbackUsuario. Es el único servicio que arranca el usuario y tiene un único método que se encarga de hacerle llegar los trinos que publican los usuarios a los que sigue, de forma automática.

IMPORTANTE: como ya se ha comentado, los trinos escritos por un determinado usuario llegan de forma automática a su lista de seguidores si están logeados en el sistema. Si un seguidor de este usuario no se encuentra logeado en ese momento en el sistema, le llegará el trino en cuanto haga login. Por tanto, hay que implementar un mecanismo en la entidad Base de datos para que guarde los trinos pendientes de enviar a un determinado usuario.

Operativa

Inicialmente la entidad Base de datos levanta su servicio Datos. Posteriormente, la entidad Servidor levanta sus dos servicios: Autenticación y Gestor.

Por último los usuarios se registran/autentican en el sistema mediante el servicio Autenticación del servidor. El usuario en cualquier momento puede decidir deslogearse (logout) del sistema y el sistema tiene que registrar ese evento convenientemente. A partir de ese momento el sistema tiene que ser capaz de almacenar los trinos que le tienen que llegar a ese usuario determinado para cuando haga login de nuevo.

Una vez que el cliente esté logeado en el sistema, podrá realizar las operaciones de mandar trinos, listar usuarios, hacerse seguidor de otros usuarios y recepción de trinos de aquellos usuarios a los que sigue. Operaciones gestionadas por la entidad Servidor.

Cuando un usuario escribe un trino, éste se imprime **directa y automáticamente** en la pantalla de todos sus seguidores. Es decir, estos no tienen que pulsar ninguna opción en ningún menú para que les aparezca el trino. El trino se imprime en pantalla en el siguiente formato:

> *nick# mensaje*

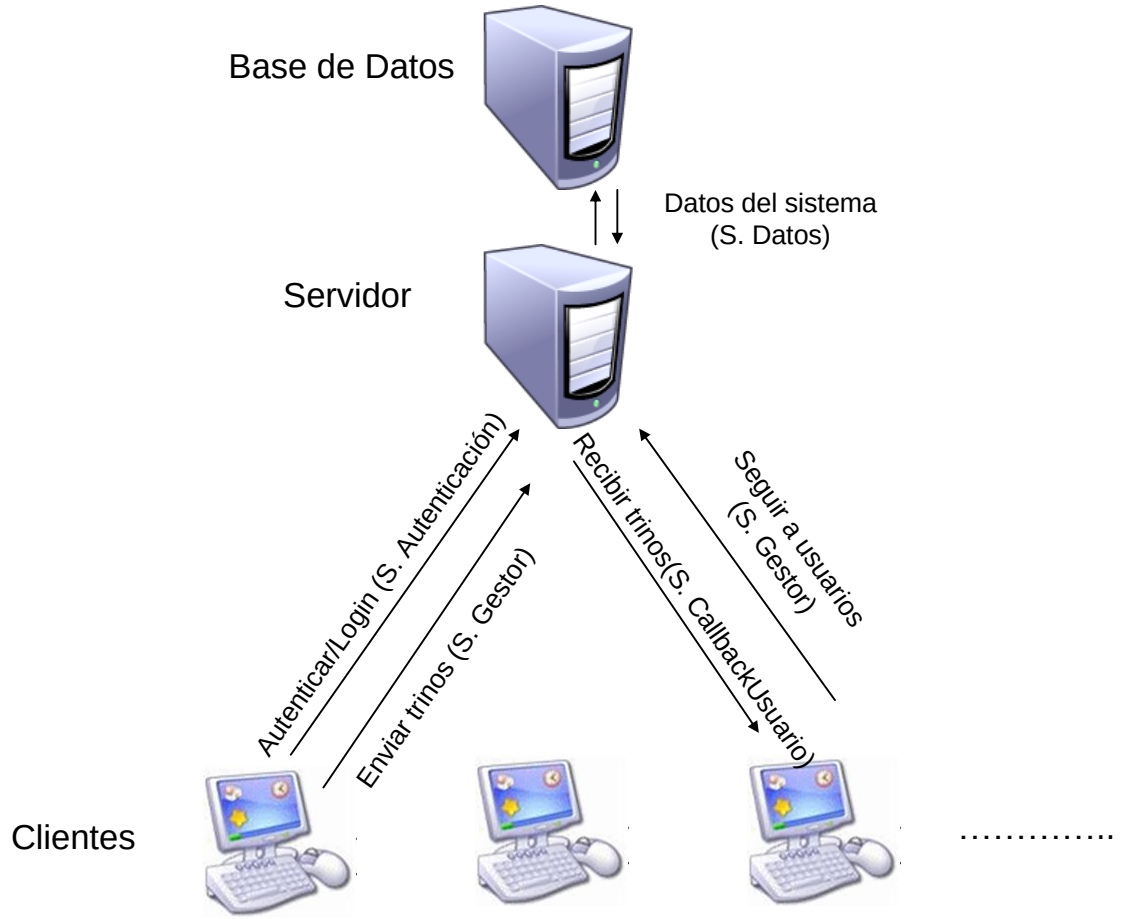
Para simplificar, la operación de hacerse seguidor de otro usuario no necesita permiso de éste. Es decir, no hace falta validar solicitudes de "amistad".

Además, cualquier usuario en cualquier momento debe tener la posibilidad de no continuar siguiendo a alguien. Para ello, el sistema debe de realizar las operaciones pertinentes para no continuar recibiendo los trinos de la persona a la que antes seguía.

Por otro lado, se propone como **tarea opcional** implementar la funcionalidad de que un usuario pueda borrar del sistema un trino, que ha sido enviado previamente, a cualquier

seguidor que no lo haya recibido todavía, por no encontrarse logeado en el sistema en el momento de su envío.

En la figura siguiente podemos observar de forma esquemática la operativa del sistema:



Interfaz

- El Servidor debe permitir mediante su interfaz de texto las siguientes operaciones:
  1.- Información del Servidor.
  2.- Listar Usuarios Logeados.
  3.- Salir.

- La Base de Datos debe permitir mediante su interfaz de texto las siguientes operaciones:
  1.- Información de la Base de Datos.
  2.- Listar Usuarios Registrados.
  3.- Listar Trinos
  4.- Salir.

- Los Clientes (usuarios) deben permitir mediante su interfaz de texto las siguientes operaciones:
  1.- Información del Usuario.
  2.- Enviar Trino.
  3.- Listar Usuarios del Sistema.

4.- Seguir a
5.- Dejar de seguir a.
6.- Borrar trino a los usuarios que todavía no lo han recibido (**opcional**).
7.- Salir "Logout".

IMPORTANTE: Cuando un usuario arranca la aplicación cliente debe aparecer inicialmente un menú con las siguientes opciones antes del menú anterior. Éste debe permitir el registro de un nuevo usuario en el sistema y/o hacer login:

1.- Registrar un nuevo usuario.
2.- Hacer login.
3.- Salir

Por favor, para facilitar la corrección de las prácticas, aténganse a presentar los menús en formato texto con las mismas opciones y orden que se le han presentado anteriormente.

Por último, como información del Servidor, Base de datos y Usuario debe aparecer en pantalla el nombre con el que se ha nombrado al/los objeto/s remoto/s que implemente el servicio que corresponda en cada caso (Ver apartado recomendaciones más abajo).

Especificaciones Generales

- Se debe utilizar código 100% Java JDK. No se admiten librerías de terceros.

- La sintaxis de llamada desde la línea de comandos tiene que ser:
  - basededatos (para iniciar el programa Base de Datos y su servicio).
  - servidor (para iniciar el programa Servidor y sus servicios).
  - usuario (para iniciar un programa cliente del usuario y su servicio).

- Por motivos de claridad, cada clase Java debe almacenarse en un fichero *.java* Todos los ficheros deben incluir, mediante comentarios al inicio de los mismos, todos los datos del autor: nombre, apellidos y correo electrónico.

- Antes de empezar a programar nada, le recomendamos que se lea varias veces esta memoria y que se haga un pequeño esquema planteando las relaciones entre cada una de las partes funcionales de la aplicación que se le pide.

- Se recomienda encapsular el código dentro de ficheros *.jar*. Esta es una forma elegante, eficiente y compacta de presentar la aplicación final.

- Pueden tomarse las decisiones de diseño que se consideren oportunas, siempre y cuando vayan en consonancia con el enunciado de la práctica y queden perfectamente comentadas en la memoria de la práctica que tiene que entregar.

- En los foros de la asignatura **no se pueden hacer preguntas del tipo**: No me funciona el programa porque me sale este error/excepción. ¿Podrían indicarme dentro de mi código fuente donde está el error o qué funciona mal?

CONSIDERACIONES DE DESARROLLO

RMI se estudia en el libro de texto básico, aunque es muy recomendable mirar el contenido de los siguientes enlaces:

-https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/index.html, página Web oficial de RMI, donde se proporciona una documentación muy amplia (en inglés) sobre la utilización de RMI y las clases asociadas.

-http://docs.oracle.com/javase/tutorial/rmi/index.html, tutorial que ayuda a entender y comprender la arquitectura de RMI. Es muy recomendable su lectura.

- Los capítulos sobre Java RMI del Libro de la bibliografía complementaria: ISBN: 978-0201796449, Título: Distributed Computing: Principles and Applications., Autor M.L. Liu., Editorial Addison Wesley Higher Education. (Existe también versión en **castellano**).

- Sesiones 5,6,7 y 8 de las tutorías Intercampus de este curso.

- Video tutorial del alumno Fermín Silva sobre Java RMI en Eclipse. http://www.youtube.com/watch?v=vnWBrCSjb44

De esta manera, se obtienen los conocimientos mínimos necesarios para instalar el entorno de desarrollo Java y ejecutar una aplicación distribuida RMI de ejemplo que sirva de base para el desarrollo de esta práctica.

Por favor, a la hora de implementar vuestra aplicación, no preguntéis en todas las opciones de los programas ¿Está seguro (s/n)? como sale en el vídeo de Fermín, ya que es muy farragoso a la hora de corregir.

Además, en el curso virtual de la asignatura se pondrá un enlace desde donde se podrá **descargar la clase Trino.java**. Esta clase Serializable es la clase base que encapsula los trinos y ya tiene implementados el constructor y los métodos de lectura de los trinos.

Detalles sobre las clases de la práctica

Con el objetivo de tener cierto orden, unificación y coherencia en el código que se entrega y para facilitar su posterior corrección, se tienen que nombrar obligatoriamente las clases/interfaces principales del programa de la siguiente manera:

`-Servidor:` Clase que contiene el *main* de la entidad Servidor.
`-Basededatos`: Clase que contiene el *main* de la entidad Base de Datos.
`-Usuario`: Clase que contiene el *main* de la entidad Usuario.
`-ServicioAutenticacionInterface`: contiene la interfaz remota del servicio de autenticación que depende de la entidad Servidor.
`-ServicioAutenticacionImpl`: clase que implementa la interfaz remota anterior.

`-ServicioGestorInterface`: contiene la interfaz remota del servicio Gestor que depende de la entidad Servidor.
`-ServicioGestorImpl`: clase que implementa la interfaz remota anterior.
`-ServicioDatosInterface`: contiene la interfaz remota del servicio Datos que depende de la entidad Base de Datos.
`-ServicioDatosImpl`: clase que implementa la interfaz remota anterior.
`-CallbackUsuarioInterface`: contiene la interfaz remota del servicio que le permite recibir al usuario los trinos de aquellos usuarios a los que sigue.
`-CallbackUsuarioImpl`: clase que implementa la interfaz remota anterior.
`-Trino`: clase que implementa y encapsula los trinos (suministrada por el equipo docente).

Aparte de éstas, se pueden utilizar todas las clases que sean necesarias pero no olvidar describir detalladamente su función en la memoria de la practica, así como, su lugar en el diagrama de clases.

RECOMENDACIONES:

-Nombrar los objetos remotos usando una *URL* que recoja toda la información sobre su dirección-puerto, servicio que presta e identificador de su proveedor; como se hace en el libro de M. L. Liu:

```
URL_nombre="rmi://"+ip+":"+rmiport+"/"+nombre_servicio+"/"+identificador_unico;
Naming.rebind(URL_nombre, objExportado);
```

- Crear una clase que encapsule todos los datos de cada usuario del sistema junto a su `CallbackUsuarioInterface`. De esta manera, se puede localizar rápidamente donde hay que remitirle los trinos en el caso de que esté logeado. (Recomendable lectura del capítulo 8 "RMI Avanzado" del libro de M. L. Liu).

- Desarrollar el código poco a poco, e ir compilando conforme se vayan obteniendo unidades funcionales.

- No es necesario cargar rmiregistry desde la línea de comandos, es posible crear una instancia del mismo mediante:

```
LocateRegistry.createRegistry(port)
```

NORMAS DE ENTREGA

Deberá entregar un único fichero comprimido en formato ZIP que contendrá una carpeta denominada PRACTICA que contendrá los ficheros fuente *.java*, los ejecutables *.class .jar* de la práctica **además de los ficheros *.bat* o *.sh* para arrancar las aplicaciones según la sintaxis de llamada comentada anteriormente**. Este fichero ZIP debe contener en su raíz un fichero PDF que contendrá la memoria de la práctica.

El fichero tiene que llamarse "Nombreestudiante_Apellidosestudiante.zip", donde Nombreestudiante es el nombre del estudiante y Apellidosestudiante son los apellidos del estudiante completos sin usar acentos y usando guiones bajos (_) en vez de espacios en blanco entre nombre y apellidos.

La memoria debe constar de los siguientes apartados:

- Portada con nombre, apellidos, DNI y correo electrónico del estudiante.
- Memoria descriptiva en la que se debe explicar el trabajo realizado. Recurra a esquemas o gráficos para plantear el funcionamiento de los programas y servicios.
- Diagramas de clases descriptivas de la estructura del sistema.
- Pantallazos de ejemplos de funcionamiento.
- Conclusiones, opiniones, y mejoras (si fuese el caso), relacionadas con la práctica.

El tamaño máximo de este documento memoria es **50 hojas** y el formato *.pdf*

La entrega de la práctica se hará a través del curso virtual y los plazos de entrega son:

- Plazo 1 (convocatoria ordinaria): prácticas recibidas entes del 15 de enero.
- Plazo 2 (convocatoria extraordinaria): prácticas recibidas con posterioridad al 15 de enero y antes del 15 de julio.