Universidad Nacional de Educación a Distancia

Grado de Ingeniería Informática

Sistema básico de almacenamiento en la nube usando Java RMI

Memoria presentada por Roberto Castellor Morant
para la asignatura Sistemas Distribuidos de Grado de Ingeniería Informática

DNI: 16041663A
Correo electrónico: roberto.castellor@gmail.com

2017

Departamento de Sistemas de Comunicación y Control

Índice general

Capítulo 1

Introducción

Este documento contiene la memoria de la práctica obligatoria de laboratorio correspondiente a la asignatura de *Sistemas Distribuidos* del Grado de Ingeniería Informática. El propósito de la practica es el desarrollo de un software que implemente un sistema de almacenamiento de ficheros en la nube usando Java RMI.

El aplicativo consiste en tres ejecutables diferentes, el servidor levanta tres servicios, dos de ellos serán usados por los distintos usuarios del servicio distribuido mientras que el tercer servicio sera para uso interno de los servicios propios del servidor.

El repositorio publica dos objetos distribuidos para uso de los clientes o del servidor, estos objetos ofrecen los métodos necesarios para almacenar los ficheros en su sistema local de ficheros o enviarlos al sistema local de ficheros de los clientes.

El cliente sera la parte utilizada por el usuario para gestionar sus ficheros, publicara a su vez un objeto distribuido que se utilizara para almacenar ficheros en el disco local del cliente.

Capítulo 2

Casos de uso

2.1. Inicio del servidor

1. El caso de uso comienza cuando el usuario lanza el proceso servidor.
2. El sistema levanta el registro para almacenar los objetos distribuidos
3. El sistema crea el objeto para el servicio de datos
4. El sistema publica en el registro RMI el servicio de datos
5. El sistema crea el objeto para el servicio de autenticación
6. El sistema publica en el registro RMI el servicio de autenticación
7. El sistema crea el objeto para el servicio gestor
8. El sistema publica en el registro RMI el servicio gestor
9. El caso de uso esta completo, el sistema entra en modo escucha de comandos

2.2. Registrar/autenticar repositorio

1. El caso de uso comienza cuando el usuario del repositorio selecciona la opción de registrar repositorio
2. El sistema solicita al usuario los datos de autenticación
3. El sistema busca el servicio de autenticación en el registro RMI
4. El sistema comprueba los parámetros de entrada con el objeto remoto
5. El sistema publica los objetos remotos para operar con el servidor en el registro
6. El sistema entra en modo consulta para operaciones con el repositorio activo

2.3. Registrar/autenticar cliente

1. El caso de uso comienza cuando el usuario del cliente selecciona la opción de registrar cliente
2. El sistema solicita al usuario los datos de autenticación
3. El sistema busca el servicio de autenticación en el registro RMI
4. El sistema comprueba los parámetros de entrada con el objeto remoto
5. El sistema publica los objetos remotos para operar con el servidor en el registro
6. El sistema entra en modo consulta para operaciones con el cliente activo

2.4. Subir fichero a la nube

1. El caso de uso comienza cuando el usuario selecciona la opción subir fichero
2. El sistema pregunta al usuario cual es el fichero que quiere subir a la nube
3. El usuario selecciona el nombre del fichero
4. El sistema crea un objeto serializable en memoria con el fichero a enviar
5. El sistema busca el objeto distribuido gestor
6. El sistema consulta al servicio gestor por la dirección del servicio operador cliente del repositorio
7. El sistema busca el objeto distribuido cliente operador del repositorio
8. El sistema envia el objeto Serializable al repositorio
9. El sistema notifica al sistema gestor el resultado del envío
10. El servicio gestor actualiza el servicio de datos con la localización del archivo
11. El sistema notifica al usuario el resultado de la operación

2.5. Descargar fichero desde la nube

1. El caso de uso comienza cuando el usuario selecciona la opcion de descargar fichero
2. El sistema pregunta al servicio gestor por los ficheros a los que tiene acceso el cliente
3. El sistema pregunta al usuario cual de los ficheros quiere descargar
4. El usuario selecciona un fichero de la lista

5. El sistema pide al servicio gestor que se envie el fichero al servicio disco cliente

2.6. Borrar fichero de la nube

1. El caso de uso comienza cuando el usuario seleccion la opción de borrar fichero
2. El sistema pregunta al servicio gestor por los ficheros del usuario
3. El sistema pregunta al usuario cual es el fichero que quiere borrar
4. El sistema pide al servicio gestor la url del objeto distribuido del servicio operador-cliente de su repositorio
5. El sistema busca en el registro RMI el servicio operador-cliente del repositorio donde esta el fichero
6. El sistema pide al servicio operador-cliente que borre el fichero
7. El sistema notifica al servicio gestor del resultado de la operación de borrado
8. El servicio gestor notifica al servicio de datos que el fichero se ha borrado fisicamente

2.7. Compartir fichero con otro cliente

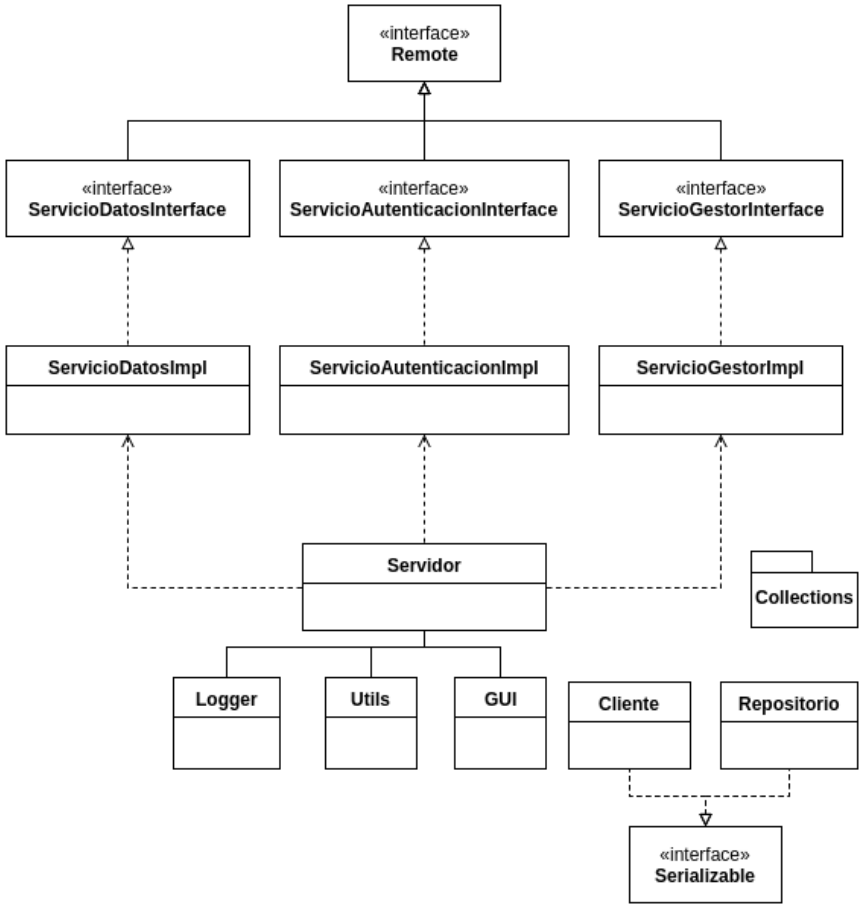
Capítulo 3

Consideraciones de diseño

3.1. Diagramas de Clases

3.1.1. Servidor

El entorno del servidor usara las siguientes clases


«interface»
Remote
«interface»
ServicioDatosInterface
«interface»
ServicioAutenticacionInterface
«interface»
ServicioGestorInterface
ServicioDatosImpl
ServicioAutenticacionImpl
ServicioGestorImpl
Servidor
Collections
Logger
Utils
GUI
Cliente
Repositorio
«interface»
Serializable

Capítulo 4

Detalles de implementación

4.1. Estructura de directorios

Las Carpeta comun contiene las clases e interfaces comunes al resto del proyecto.

- **Cliente**, clase con la información de los clientes, el usuario, el id y listas con los ficheros del cliente
- **Fichero**, clase proporcionada por el equipo docente para enviar los ficheros
- **GUI**, clase con métodos genéricos para mostrar por pantalla información y recoger las entradas del usuario
- **Logger**, clase con métodos para volcar a un fichero log de la aplicación y de esta forma no mezclarse con la interfaz del usuario
- **Metadato**, clase con información referente a los ficheros, tendrá el repositorio, el id del cliente propietario del fichero, etc.
- **Repositorio**, clase con datos de los repositorios registrados en el sistema, los id, el usuario y las url de los objetos remotos que publica el repositorio.
- *ServicioAutenticacionInterface*, interfaz del servicio de autenticación.
- *ServicioDatosInterface*, interfaz del servicio de datos
- *ServicioGestorInterface*, interfaz del servicio gestor
- *ServicioClOperadorInterface*, interfaz del servicio cliente-operador
- *ServicioSrOperadorInterface*, interfaz del servicio servidor-operador
- *ServicioDiscoClienteInterface*, interfaz del servicio de disco del cliente
- **Utils**, clase con métodos genéricos para el uso por parte de la aplicación

La carpeta servidor, contiene el código necesario para lanzar el servidor y la implementación del servicio de datos, el servicio de autenticación y el servicio gestor.

- **Servidor**, clase que lanza el servidor, crea los objetos remotos y los publica en el registro RMI
- **ServicioAutenticacionImpl**, clase que implementa el servicio de autenticación
- **ServicioDatosImpl**, clase que implementa el servicio de datos
- **ServicioGestorImpl**, clase que implementa el servicio gestor

La carpeta repositorio contiene el código necesario para lanzar los repositorios y la implementación de los servicios operador-cliente y operador-servidor.

- **Repositorio**, clase que lanza el repositorio y genera el menú para operar con el, publica los objetos remotos.
- **ServicioClOperadorImpl**, clase que implementa el servicio cliente-operador
- **ServicioSrOperadorImpl**, clase que implementa el servicio servidor-operador

La carpeta cliente contiene el código necesario para lanzar los clientes y la implementación de la interfaz remota del servicio de disco del cliente

- **Cliente**, clase que lanza el cliente, genera el menú para poder operar con el y publica los objetos remotos
- **ServicioDiscoClienteImpl**, clase que implementa el servicio de disco del cliente a través del cual se recibirán ficheros de la nube

4.2. Modelo de datos

El modelo de datos esta basado en la api Collections de Java.

Cada elemento del sistema sera identificado por un id numérico que formara parte del objeto como atributo.

En el servicio de datos se almacenara la siguiente información

Lista con todos los clientes registrados en la aplicación. Relación de nombre de usuario con id de clientes con datos de cliente en una HashMap¡String, Integer¿de la que se podrá obtener el id de un cliente dado su nombre de usuario. Relación de id de usuario con el objeto de tipo Cliente que tiene toda la información referente a cada cliente. Variable sesioncli con el ultimo identificador de cliente asignado, a

cada cliente nuevo se le asignara como identificador el valor de esta variable y esta se incrementara en espera del próximo cliente.

La clase Cliente del paquete comun tiene la información propia del cliente y dos mapas indexadas por el nombre del fichero, una con los metadatos de los ficheros cuyo propietario es el cliente y otra con los metadatos de los ficheros que se han compartido con el cliente. Las operaciones con los datos de los ficheros se realizaran mediante llamadas a metodos de esta clase y en ningún momento accediendo directamente a las collection que albergan los datos.

Capítulo 5

Ejemplos de uso

En este capitulo mostraremos ejemplos de uso de la aplicación, para los ejemplos he registrado en el servidor cuatro repositorios y siete clientes, cada cliente tendra asociados varios ficheros.

5.1. Operaciones del servidor

```
==========Menu Servidor==========
1-Listar Clientes
2-Listar Repositorios
3-Listar Parejas Repositorio-Cliente
4-Salir
==========Menu Servidor==========
Seleccione una opcion: 1
Cliente: 0-cliente1
Cliente: 1-cliente2
Cliente: 2-cliente3
Cliente: 3-cliente4
Cliente: 4-cliente5
Cliente: 5-cliente6
Cliente: 6-cliente7
==========Menu Servidor==========
1-Listar Clientes
```

```
2-Listar Repositorios
3-Listar Parejas Repositorio-Cliente
4-Salir
==========Menu Servidor==========
Seleccione una opcion: 2
-- Repositorios registrados en el sistema:
Repositorio: 0-repositorio1
Repositorio: 1-repositorio2
Repositorio: 2-repositorio3
Repositorio: 3-repositorio4
==========Menu Servidor==========
1-Listar Clientes
2-Listar Repositorios
3-Listar Parejas Repositorio-Cliente
4-Salir
==========Menu Servidor==========
Seleccione una opcion: 3
Cliente: 0-cliente1 <->0-repositorio1
Cliente: 1-cliente2 <->1-repositorio2
Cliente: 2-cliente3 <->2-repositorio3
Cliente: 3-cliente4 <->3-repositorio4
Cliente: 4-cliente5 <->0-repositorio1
Cliente: 5-cliente6 <->1-repositorio2
Cliente: 6-cliente7 <->2-repositorio3
==========Menu Servidor==========
1-Listar Clientes
2-Listar Repositorios
3-Listar Parejas Repositorio-Cliente
4-Salir
==========Menu Servidor==========
Seleccione una opcion:
```

Apéndice A

Herramientas

Para realizar esta practica se han utilizado estas herramientas:

OpenJDK Java 1.8 http://openjdk.java.net/ Ant para construir los ejecutables del proyecto http://ant.apache.org/ Eclipse para edición del código http://www.eclipse.org/ git como sistema para mantener el control de la evolución del código y permitir desarrollar desde varios PC https://git-scm.com/ draw.io, herramienta en la nube para generar graficos https://www.draw.io/ Latex, entorno tex para realizar la memoria