

# Sistemas distribuidos

**Miroslav Vladimirov**

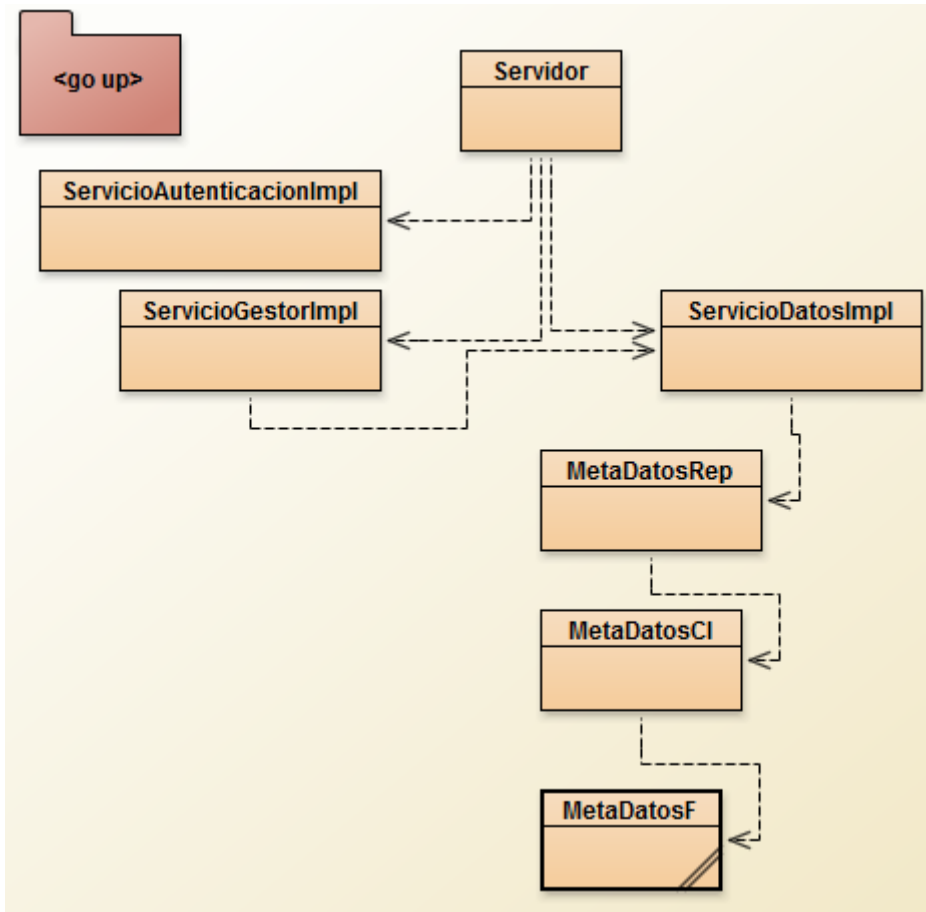
**Email:** [miro.kv89@gmail.com](mailto:miro.kv89@gmail.com)

**Sistema básico de almacenamiento en la nube  
usando Java RMI**

El propósito de la práctica es el desarrollo de un software que implemente un sistema de almacenamiento de ficheros en la nube usando Java RMI.

En nuestro sistema interactúan tres tipos de actores que describimos a continuación:

## 1. Servidor



Al iniciar el servidor éste generará una dirección URL en la que será posible localizarlo:

<rmi://localhost:7000/Servidor>

Tanto el cliente como el repositorio utilizarán dicha dirección para ponerse en contacto con el servidor y pedir las direcciones de sus servicios.

A continuación, el servidor pondrá en marcha sus servicios, les proporcionará la base de datos y aparecerá en pantalla el menú con las distintas opciones:

```
C:\Users\Wifi\workspace\SIDI>java -jar servidor.jar
Servidor en funcionamiento.
=== Menu Servidor ===
Seleccione la opcion.

1.- Listar clientes
2.- Listar Repositorios
3.- Listar Parejas Repositorio-cliente
4.- Salir
```

El servidor es el encargado de controlar el proceso de almacenamiento de ficheros y gestionar los recursos de almacenaje, para ello hace uso de tres servicios:

- El servicio de autenticación se encarga tanto del registro como de la autenticación de los clientes y repositorios. El funcionamiento de este servicio consiste en exportar un objeto remoto con una URL específica:

<rmi://localhost:8000/ServicioAutenticacion>

Tras lo cual se mantiene a la espera de que los usuarios hagan uso de él.

- El servicio Gestor se encarga de gestionar las operaciones de los clientes en relación a sus ficheros. Usando este servicio el cliente podrá subir, bajar, borrar, listar y compartir sus ficheros. El funcionamiento de este servicio consiste en exportar un objeto remoto con una URL específica:

<rmi://localhost:9000/ServicioGestor>

tras lo cual se mantiene a la espera de que los usuarios hagan uso de él.

- El servicio Datos es el encargado de gestionar la base de datos. Se ha estructurado de tal forma que la única lista que contiene es de MetaDatosRep, esto son los datos de los repositorios, que a su vez contienen una lista MetaDatosCl que son los datos de los clientes, y por último, estos contienen una lista de MetaDatosF que son los datos de los ficheros de esos clientes. Se ha elegido esta forma de hacerlo puesto que no puede existir un fichero que no esté asignado a un cliente, así como tampoco puede haber un cliente registrado que no tenga asociado un repositorio.

Los dos servicios anteriores hacen uso de éste. Se ha optado por no poner dicho servicio remotamente, sino, que sea el servidor quien lo proporcione a sus servicios.

Inicialmente éste servicio importa el archivo “basedatos” que es el que contiene la lista de MetaDatosRep, y después con cada modificación de la base de datos, dicho archivo se sobrescribe, por lo que no se guardan versiones anteriores (si un archivo es borrado, es irrecuperable).

A continuación, se muestran las distintas operaciones que puede realizar el servidor:

Listar Clientes:

```
=== Menu Servidor ===
Seleccione la opcion.

1.- Listar clientes
2.- Listar Repositorios
3.- Listar Parejas Repositorio-cliente
4.- Salir
1

Listar clientes
Nombre: Fecha de registro:
miro 27-nov-2016 16:40:12
```

Listar Repositorios:

```
=== Menu Servidor ===
Seleccione la opcion.

1.- Listar clientes
2.- Listar Repositorios
3.- Listar Parejas Repositorio-cliente
4.- Salir
2

Lista de Repositorios:

Nombre          Fecha registro:
repo            27-nov-2016 16:40:01
```

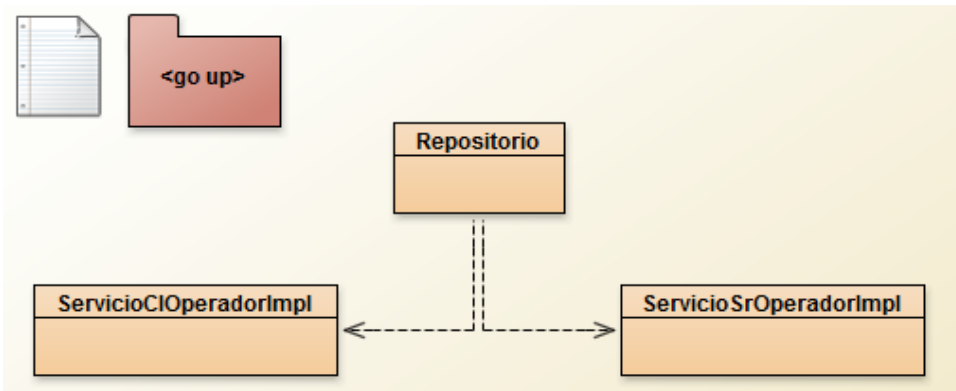
Listar Parejas Repositorio-Cliente:

```
=== Menu Servidor ===
Seleccione la opcion.

1.- Listar clientes
2.- Listar Repositorios
3.- Listar Parejas Repositorio-cliente
4.- Salir
3

Listar Parejas Repositorio-cliente
Repositorio: repo      Activo desde:27-nov-2016 16:40:01
Clientes:
Nombre: Fecha de registro:
miro    27-nov-2016 16:40:12
```

## 2. Repositorios



Estas entidades son las responsables de guardar en sus dispositivos de almacenamiento los ficheros que los clientes suben a la nube. Cuando un cliente solicita la subida/bajada/borrado de un fichero a través del servicio Gestor del servidor, este servicio Gestor accede a la base de datos en busca del repositorio que corresponde a dicho cliente para finalmente devolver la información necesaria para completar la operación. Al arrancar un repositorio pone en marcha sus dos servicios y muestra por pantalla el siguiente menú:

```
C:\Users\Wifi\workspace\SIDI>java -jar repositorio.jar
=== Menu principal Repositorios ===
Seleccione la opcion.

1.- Autenticarse
2.- Registro nuevo
3.- Salir
```

- El servicio cliente-Operador se encarga de las operaciones de subida y borrado de ficheros. Cuando un cliente solicita una de las dos operaciones al servicio Gestor del servidor, éste le responde con la URL del servicio cliente operador del repositorio que tiene asignado. Después el cliente llama a este servicio que se encargará de realizar las operaciones solicitadas. Para acceder a este servicio utilizaremos la siguiente dirección URL:

<rmi://localhost:11000/ServicioCOp>

- El servicio Servidor-Operador se encarga de las descargas desde el repositorio hacia el cliente. Para realizar dicha operación el cliente solicita al servicio Gestor del servidor la descarga de un fichero, el servicio Gestor llama al servicio Servidor-Operador pasándole la URL del servicio DiscoCliente que es el que se encarga de la descarga. Para acceder a este servicio utilizaremos la siguiente dirección URL:

<rmi://localhost:11001/ServicioSrOp>

Una vez autenticado el repositorio se muestra el siguiente menú con las distintas operaciones que puede realizar:

```
=== Menu Repositorio ===
Seleccione la opcion.

1.- Listar Clientes
2.- Listar ficheros del Cliente
3.- Salir
```

Puesto que confiamos en que el uso responsable de los repositorios, he decidido que no sea necesaria una contraseña para su acceso. A continuación, se muestran las operaciones posibles:

Listar Clientes del repositorio:

```
=== Menu Repositorio ===
Seleccione la opcion.

1.- Listar Clientes
2.- Listar ficheros del Cliente
3.- Salir
1

Listar Clientes
Usuarios conectados:
Nombre: Fecha de registro:
miro 27-nov-2016 16:40:12
```

Listar ficheros de un cliente:

```
=== Menu Repositorio ===
Seleccione la opcion.

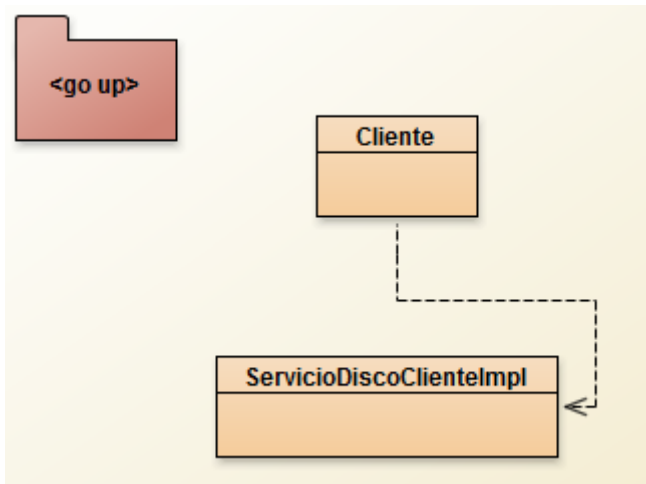
1.- Listar Clientes
2.- Listar ficheros del Cliente
3.- Salir
2

Listar ficheros del Cliente
Escriba el ID del Cliente:miro

Ficheros de: miro

Nombre      Propietario      Fecha de subida:
as.txt      miro             27-nov-2016 16:40:28
```

### 3. Clientes



Los clientes son los propietarios de los ficheros. Se registran a través del servicio de autenticación del servidor para poder gestionar sus ficheros a través del servicio Gestor. Cada cliente publica la interfaz de su servicio DiscoCliente, que será utilizado para la descarga de ficheros en el disco duro local del cliente. Una vez arrancado el cliente.jar, se muestra por pantalla el siguiente menú:

```
C:\Users\Wifi\workspace\SIDI>java -jar cliente.jar
=== Menu principal ===
Seleccione la opcion.

1.- Autenticarse
2.- Registro nuevo
3.- Salir
```

Después se queda a la espera de que un usuario se registre o autentique. Una vez realizado el registro se le asigna un repositorio al nuevo cliente. Cuando entra en el sistema autenticándose, se pone en marcha su servicio DiscoCliente al que accederemos con la siguiente dirección URL:

<rmi://localhost:15000/ID del cliente/ServicioDC>

Después se muestra el menú con las operaciones que puede realizar el cliente:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
```

A continuación, se muestran las distintas operaciones que se pueden realizar:

Subida de un fichero:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
1

Escriba la ruta del fichero que desea subir:E:/SIDI
Escriba el nombre del fichero que desea subir:Pr.pdf
Archivo Pr.pdf subido con exito.
```

Descarga de un fichero:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
2

Escriba el nombre del fichero que desea descargar:Pr.pdf
Si el fichero es suyo pulse intro, en caso contrario escriba el nombre del cliente al que pertenece
La descarga se ha completado
```

Borrado de un fichero:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
3

Escriba el nombre del fichero que desea eliminar:Pr.pdf
Archivo "Pr.pdf" borrado con exito.
```

5

Compartir Fichero:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
4

Escriba el nombre del fichero que desea compartir:Pr.pdf
```



Listar Ficheros:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
5

Listar Ficheros
Mis Ficheros:

Nombre          Propietario      Fecha de subida:
Pr.pdf          miro            27-nov-2016 16:40:28

=== FICHEROS COMPARTIDOS ===

Nombre          Propietario      Fecha de subida:
Pr.pdf          miro            27-nov-2016 16:40:28
```

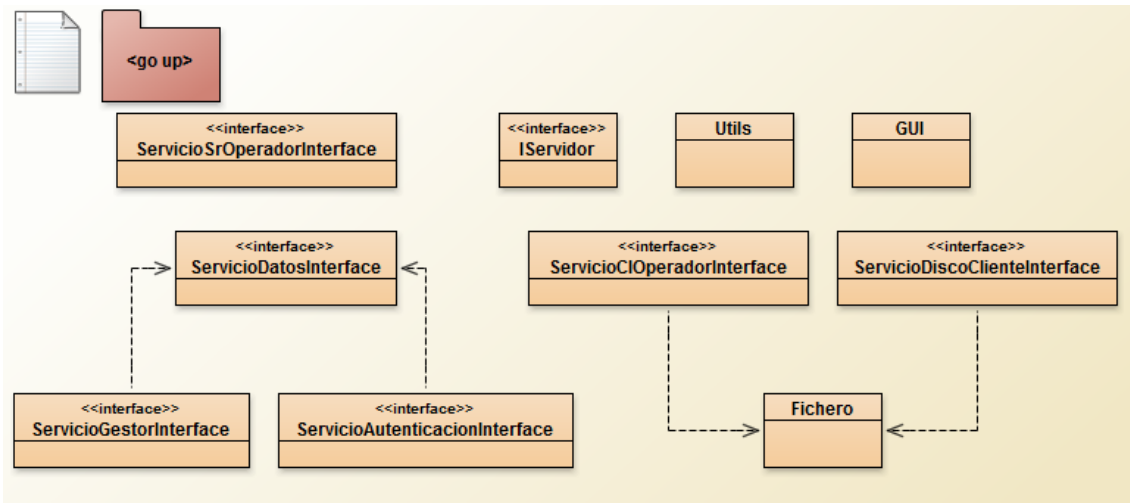
Mostrar lista de clientes:

```
=== Menu Usuario ===
Seleccione la opcion.

1.- Subir ficheros
2.- Bajar ficheros
3.- Borrar ficheros
4.- Compartir ficheros
5.- Listar ficheros
6.- Mostrar lista de clientes
7.- Salir
6

Usuarios conectados:
Nombre: Fecha de registro:
miro    27-nov-2016 16:40:12
```

## 4. Common



En el paquete Common tenemos todas las interfaces, así como el `Fichero.java` proporcionado por el equipo docente, una clase `Utils`, y otra `GUI` que utilizaremos en los tres paquetes anteriores. Al exportar los ejecutables \*.jar se ha incluido el paquete common en cada uno de ellos. Como se puede observar, tanto el servicio de autenticación como el servicio gestor hacen uso del servicio de datos. Por otro lado, también vemos que los servicios Cliente-Operador y DiscoCliente hacen uso de `Fichero.java`, como era de esperar ya que son los encargados de subir/bajar/borrar ficheros.

## 5. Conclusiones y opinión personal:

La práctica ha resultado ser mucho más complicada de lo esperado. La mayor dificultad no ha sido comprender el funcionamiento de Java RMI, como se esperaba al principio, sino comprender los requerimientos del enunciado, que han resultado de lo más confusos. Se han encontrado fallos en el enunciado, que luego han sido aclarados por el Equipo Docente en los foros de la asignatura como el hecho de que en el enunciado pone que la carpeta del cliente que se debe crear en el repositorio se crea en el momento de la autenticación, en cambio lo correcto es crearla en el momento del registro.

Tras acabar la práctica y encontrarme con varios errores de este tipo, y los que yo mismo he generado al interpretar el enunciado de otra forma, me he visto en el problema de tener que rehacer la práctica casi de cero.

Por lo demás, ha sido divertido, Java es un lenguaje sencillo y hacer un “Dropbox casero” es un reto en el que apetece invertir tiempo.

Me gustaría que se mejoraran cosas en cuanto al enunciado, podría hacerse mucho más corto y comprensible si se proporcionase una arquitectura ya creada con las clases existentes y su distribución. Los caminos a seguir para la subida/bajada/borrado de ficheros son muy confusos si en todas ellas pone que debe pasar por el servidor (lo que se comprende como la clase `Servidor.java`), cuando se refieren a alguno de sus servicios en realidad.