

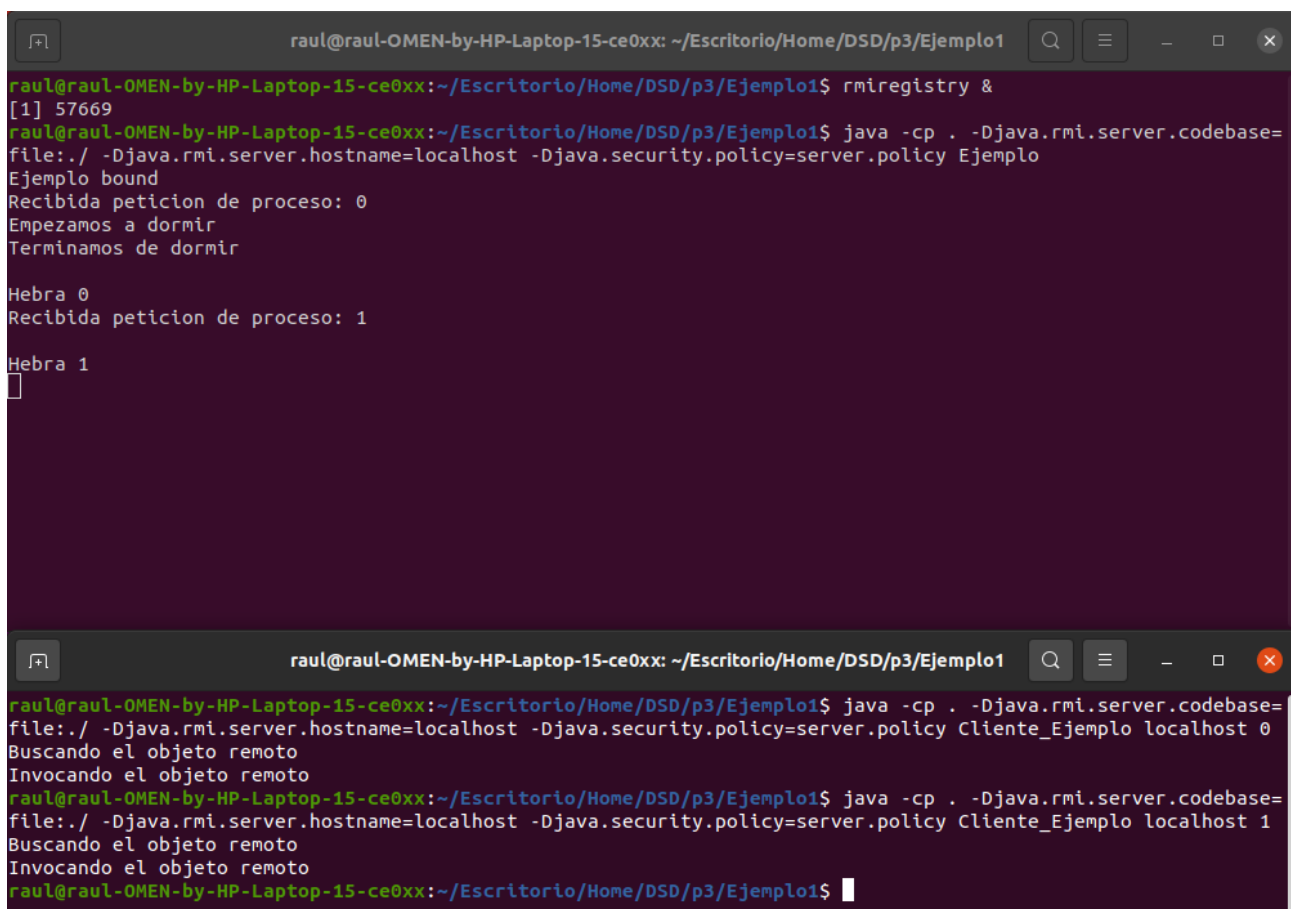
Documentación P3 – RMI

Raúl Castro Moreno
3º IS – DSD2

1. Ejemplos

1.1 Primer ejemplo

En este ejemplo, lo único que tenemos es un método el cuál nos devuelve el número que le hemos pasado con el mensaje “Hebra X”, sin embargo, si le pasamos el 0, antes de devolvernos el mensaje, duerme durante 5 segundos. Esta basado en lanzar varios clientes con diferentes identificadores de proceso, para mostrar que es concurrente RMI.



```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Ejemplo1
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Ejemplo1$ rmiregistry &
[1] 57669
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Ejemplo1$ java -cp . -Djava.rmi.server.codebase=
file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Ejemplo
Ejemplo bound
Recibida peticion de proceso: 0
Empezamos a dormir
Terminamos de dormir

Hebra 0
Recibida peticion de proceso: 1

Hebra 1
[1]

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Ejemplo1
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Ejemplo1$ java -cp . -Djava.rmi.server.codebase=
file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo localhost 0
Buscando el objeto remoto
Invocando el objeto remoto
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Ejemplo1$ java -cp . -Djava.rmi.server.codebase=
file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo localhost 1
Buscando el objeto remoto
Invocando el objeto remoto
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Ejemplo1$
```

1.2 Segundo ejemplo

Ahora en vez de tener que lanzar varios clientes, el mismo cliente lanza varias hebras para realizar la operación de escribir_mensaje. El parámetro que le pasamos ahora es el número de hebras que va a lanzar el cliente. En caso de ser 0, no se llegan a lanzar puesto que el vector de hebras se inicializa con ese valor, y lo que realiza el método start() de las hebras, es un bucle for que va de 0 al tamaño del vector -1, y como en este caso seria de 0 a 0 -1, ni se llega a entrar en el bucle.

Si al método de escribir mensaje le ponemos synchronized, ocurre que, ese método tendría como un semáforo, el cuál bloquea a los demás threads hasta que este no termine de realizar el método. En la ejecución sin synchronized, se observa como todos los threads entran a ejecutar el mismo método a la vez, mientras que el de synchronized, hasta que una hebra no termina de ejecutarlo, otra no empieza a ejecutarlo.

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Ejemplo
Ejemplo bound

Entra Hebra Cliente 4
Entra Hebra Cliente 3
Entra Hebra Cliente 2
Entra Hebra Cliente 0
Empezamos a dormir
Sale Hebra Cliente 3
Sale Hebra Cliente 4
Sale Hebra Cliente 2

Entra Hebra Cliente 1
Sale Hebra Cliente 1
Terminamos de dormir
Sale Hebra Cliente 0
█

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo_Mul
ti_Threaded localhost 5
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ █
```

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Ejemplo
Ejemplo bound
█

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo_Mul
ti_Threaded localhost 0
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ █
```

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Ejemplo
Ejemplo bound

Entra Hebra Cliente 2
Sale Hebra Cliente 2

Entra Hebra Cliente 1
Sale Hebra Cliente 1

Entra Hebra Cliente 4
Sale Hebra Cliente 4

Entra Hebra Cliente 3
Sale Hebra Cliente 3

Entra Hebra Cliente 0
Empezamos a dormir
Terminamos de dormir
Sale Hebra Cliente 0

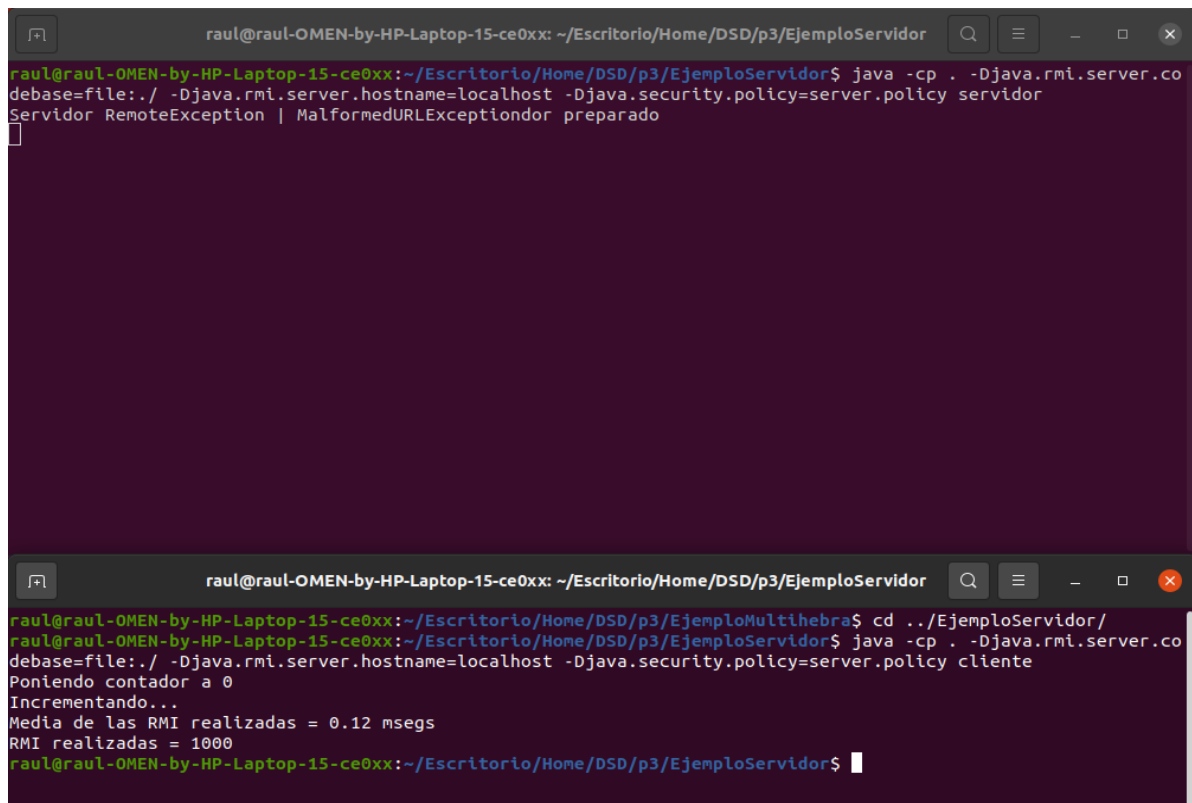
```

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploMultihebra
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ java -cp . -Djava.rmi.server.
codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente_Ejemplo_Mul
ti_Threaded localhost 5
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Buscando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
Invocando el objeto remoto
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$
```

1.3 Tercer ejemplo

En este ejemplo, tenemos un servidor, que lo que hace es crear un instancia de la clase contador, la cual implementa la interfaz remota de icontador, y a esa instancia, le realiza un rebind, para ponerle un nombre por el cuál localizarla en el registro y obtener su referencia para poder realizar las operaciones. Esta se obtiene, haciendole al registro, un lookup con el nombre el cuál ha puesto el servidor al usar rebind. Ya con la referencia, se le pide que realice su operación de incrementar 1000 veces, y calcula el tiempo en el que tarda en hacerlo, y nos lo muestra por pantalla.

Tengo que comentar, que no me funcionaba este ejemplo debido a que tenía la orden de **rmiregistry** & activada, teniendo así el puerto 1099 ocupado. Por tanto he tenido que buscarlo y matarlo mediante la orden **kill -9 PID**.



```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploServidor
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploServidor$ java -cp . -Djava.rmi.server.co
debase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy servidor
Servidor RemoteException | MalformedURLExceptionondor preparado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/EjemploServidor
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploMultihebra$ cd ../EjemploServidor/
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploServidor$ java -cp . -Djava.rmi.server.co
debase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy cliente
Poniendo contador a 0
Incrementando...
Media de las RMI realizadas = 0.12 msecs
RMI realizadas = 1000
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/EjemploServidor$
```

2. Servidores Replicados

2.1 Descripción

Para realizar este ejercicio he utilizado solo 3 archivos “.java”, los cuales son **I_Donaciones**, **Cliente** y **Servidor**.

I_Donaciones es la interfaz remota.

Servidor extiende **I_Donaciones** y realiza todas las implementaciones de los métodos de la interfaz, además, aparte de esos métodos, tiene uno más el cuál se llama **enlazar()** y es el que utiliza para conectar los servidores entre sí y poder usar las réplicas.

Cliente se comunica con el Servidor, pidiéndole que realice las operaciones que este ofrece.

2.2 Métodos Servidor y Cliente

Servidor contiene lo más importante:

Atributos:

- **usuarios** → Array de tipo **String** que guarda los **nombres de los usuarios registrados**.
- **totalLocal** → Variable de tipo **double**, que **guarda el dinero donado**.
- **replica** → Variable de tipo **I_Donaciones**, que guarda la **referencia a la réplica de Servidor que se va a usar**.

- **v_replicas** → Array de tipo **I_Donaciones**, que guarda las **referencias de todas las réplicas** de servidores, incluyendo la suya misma.
- **registro** → Array de tipo **String**, que guarda el **nombre de los usuarios que han donado** una vez mínimo.

Métodos:

- **registrar(String usuario)** → Añade al array de **usuarios** de la réplica con menos usuarios registrados, al usuario que se le pasa por parámetro, solo si no figura en ningún array de usuarios de las distintas replicas, incluida ella misma.
- **donar(String usuario, double cantidad)** → Comprueba si el usuario pasado por parámetro figura registrado en alguna de las réplicas, incluida ella misma, en caso afirmativo, en la réplica la cual este el usuario registrado, se le suma cantidad a la variable **totalLocal**. Además, se añade a la variable **registro** indicando que ha donado.
- **getTotal(String usuario)** → Devuelve el **total donado**, el cual es la **suma de las variables totalLocal de todas las réplicas**, solo en el caso, de que el usuario que se pasa por parámetro, **esté registrado**, y que su nombre figure en el array de **registro**, es decir, que haya donado al menos una vez.
- **addReplica(I_Donaciones ref)** → Añade la referencia que se le pasa por parámetro, al array de **v_replicas**.
- **addUsuario(String usuario)** → Añade el usuario pasado por parámetro al vector de **usuarios** registrados.
- **addDonante(String donante)** → Añade al array **registro** el valor pasado por parámetro.
- **donacion(double cantidad)** → Suma a la variable totalLocal el valor pasado por parámetro.
- **getUsuarios()** → Devuelve el array de **usuarios**.
- **getUsuariosTotales()** → Devuelve un **array con todos los usuarios**, incluidos los que están en otras réplicas registrados.
- **getRegistro()** → Devuelve el array de **registro**.
- **getTotalLocal()** → Devuelve la variable **totalLocal**.
- **getReplicas()** → Devuelve el array de **v_replicas**.
- **enlazar(string ubicación, string nombre)** → Este método, es el que realiza la conexión entre réplicas, para ello le pasamos dos strings, uno de la ubicación donde están los servidores(en mi caso localhost) y otro del nombre del servidor original. Empieza creando un **registro** localizado en la **ubicación**, y de ese registro obtenemos una referencia al servidor original utilizando el **nombre** con el método **lookup**.
Ahora, obtengo el vector de réplicas del servidor original, y compruebo:

Si el tamaño del vector es 0, no he hecho antes todavía ninguna réplica, por tanto, al **v_replicas** de el servidor que estoy replicando, le añado la referencia del servidor original y la suya misma, y al servidor original, también le añado a su **v_replicas**, la referencia al nuevo servidor, y la referencia así mismo, en este caso al servidor original. De esta manera los 2 servidores, tienen referencias así mismos y al otro.

Si el tamaño del vector es mayor o igual que 2, significa entonces que ya tenemos mínimo 2 réplicas, el original y su réplica. Por tanto, lo que vamos a hacer es iterar por el vector de réplicas del servidor original obtenido antes, y cada una de las referencias que tiene ese vector, le añadimos a su vector de réplicas la nueva referencia, es decir, la del servidor replicado nuevo con el cual estamos llamando al método **enlazar**, y además, esa misma referencia a la cual le hemos añadido la nueva, la usamos para añadirla al **v_replicas** del servidor nuevo. De esta manera, todos los servidores, añaden a su vector de replicas, la referencia a la nueva replica, y esta nueva replica, añade a su vector, las referencias de todas las replicas que había antes, y por último le añade la suya misma.

El Cliente solo tiene un método main, que a parte de realizar la conexión con el servidor remoto que le indiquemos por parámetro, tiene un switch, con el cual podemos elegir las operaciones que queremos realizar.

2.3 Colaboración de servidores

Para la conexión entre servidores replicados, hemos utilizado lo que viene explicado en el método **enlazar()** del **Servidor**. Se basa , en que cada servidor tiene referencias a todas las demás réplicas, para poder comprobar datos(en este caso, usuarios registrados, dinero donado, etc).

2.4 Extensión de funcionalidad

Hemos hecho 2 cosas para extender la funcionalidad:

- **Poder usar más de 2 réplicas**, todo el programa esta diseñado para usar réplicas infinitas, sin embargo, como para crear los servidores hacen falta los nombres, lo he limitado a **5** réplicas.

- Aparte de registrarse, donar y consultar el total donado , **hemos añadido más funcionalidades**, como la de **obtener el total local donado**, **obtener los usuarios registrados localmente** y **los usuarios totales registrados entre todas las réplicas**.

2.5 Ejemplo de ejecución.

Para ejecutar todo he usado la línea de comandos.

Antes de los siguientes comandos, compilo todo con **javac *.java**, y se crean los archivos **“.class”**. En esa misma carpeta donde se crean, pongo el comando: **rmiregistry &**

La orden para el **Servidor** es la siguiente:

```
java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor (IP) (Servidor)
```

Hay que poner la IP y el Servidor en la orden

MUY IMPORTANTE → el primer servidor que se inicie debe ser el 0, para marcar el servidor original.

Ejemplo:

java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0

La orden para el **Cliente** es la siguiente:

java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente (IP) (Servidor)

Hay que poner la IP y el Servidor en la orden (el servidor que marquemos debe estar activo)

Ejemplo:

java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 2

Para el ejemplo de ejecución, vamos a levantar 3 servidores, y utilizar sus funcionalidades, incluyendo las que dan errores, tener en cuenta que en todas las fotos, no se han reiniciado los servidores, solo cambiado el servidor con el que interactúa el cliente.

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/ServidoresReplicados$ javac *.java
raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0
Servidor 0 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 1
2 servidores enlazados correctamente
Servidor 1 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx:~/Escritorio/Home/DSD/p3/Servidor...$ java -cp . -Djava.rmi.server.codebase=file:./ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 2

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
0

Introduzca nombre de usuario a registrar
Pepe

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
1

Introduzca nombre de usuario y cantidad a donar
Pepe
300

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
2

Introduzca nombre de usuario
Pepe
Total donado(0 = No puedes consultarlo): 300.0

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
```

Usando función de Registrar, donar, consultar total donado en el servidor 2.

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0
Servidor 0 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 1
3 servidores enlazados correctamente
Servidor 1 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Servidor...$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 2

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

3
Total donado localmente(0 = No puedes consultarlo): 300.0

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

4
Usuarios registrados localmente: Pepe

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

5
Usuarios registrados: Pepe

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
```

Consultando lo donado localmente, usuarios registrados localmente y usuarios registrados totales en el servidor 2.

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0
Servidor 0 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 1
2 servidores enlazados correctamente
Servidor 1 se ha iniciado
El usuario Pepe ya esta registrado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Servidor...$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 1

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

0
Introduzca nombre de usuario a registrar
Pepe

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

0
Introduzca nombre de usuario a registrar
Jose

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

0
Introduzca nombre de usuario a registrar
Javi

Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir

4
Usuarios registrados localmente: Jose
```

Probando a registrar un usuario ya registrado desde el servidor 1, el fallo aparece en el servidor 1. Registramos más usuarios , pero al comprobar los usuarios locales, el segundo no aparece, puesto que se ha registrado en la réplica que tenía menos usuarios registrados(servidor 0)


```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ javac *.java
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0
Servidor 0 se ha iniciado
No puedes donar sin estar registrado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 1
2 servidores enlazados correctamente
Servidor 1 se ha iniciado
El usuario Pepe ya esta registrado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor iniciado
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor iniciado
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Servidor...$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 0
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
1
Introduzca nombre de usuario y cantidad a donar
Marcos
400
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
1
Introduzca nombre de usuario y cantidad a donar
Jose
400
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
2
Introduzca nombre de usuario
Jose
Total donado(0 = No puedes consultarlo): 700.0
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
5
Usuarios registrados: Javi Jose Pepe
```

Ahora desde el servidor 0, intentamos donar con un usuario no registrado, aparece el fallo en el servidor 0, donamos con uno si registrado y consultamos el total donado. La suma de lo donado hasta ahora esta correcta. Miramos también los usuarios totales registrados, y aparecen correctamente todos.

```
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ javac *.java
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 0
Servidor 0 se ha iniciado
No puedes donar sin estar registrado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 1
2 servidores enlazados correctamente
Servidor 1 se ha iniciado
El usuario Pepe ya esta registrado

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor iniciado
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor iniciado
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Servidor localhost 2
3 servidores enlazados correctamente
Servidor 2 se ha iniciado
El usuario no puede consultar el total donado sin estar registrado antes
El usuario no puede consultar el total donado hasta que realice como minimo una donación

raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/Servidor...$ java -cp . -Djava.rmi.server.codebase=file:/ -Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy Cliente localhost 2
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
2
Introduzca nombre de usuario
Jorge
Total donado(0 = No puedes consultarlo): 0.0
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
2
Introduzca nombre de usuario
Javi
Total donado(0 = No puedes consultarlo): 0.0
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
3
Total donado localmente(0 = No puedes consultarlo): 300.0
Elija que desea hacer
0 - Registrarse
1 - Donar
2 - Consultar total donado
3 - Consultar total local
4 - Consultar usuarios locales
5 - Consultar usuarios globales
6 - Salir
6
Pase buen dia
raul@raul-OMEN-by-HP-Laptop-15-ce0xx: ~/Escritorio/Home/DSD/p3/ServidoresReplicados$
```

Desde el servidor 2 , intentamos consultar el total con un usuario no registrado, y aparece el fallo en el servidor, lo mismo pero con un usuario registrado, pero que no ha donado ni una vez, y sale el fallo. Por último, comprobamos el total local que hemos donado al principio, y aparece correctamente.