

## P3: Lenguajes y Middlewares para Programación Distribuida

Salvador Romero Cortés

### Ejemplos

- **Ejemplo 1:** Cuando se introduce un 0, el servidor se duerme durante 5 segundos
- **Ejemplo 2:** Cuando llega al servidor una hebra cuyo nombre acaba en 0 el servidor espera durante 5 segundos. La diferencia entre poner `synchronized` y no ponerlo es que si no lo ponemos el resto de hebras seguirán escribiendo mensajes en la terminal mientras duerme la hebra que acaba en 0. Si le ponemos `synchronized` el resto de hebras no escribirán en pantalla hasta que la hebra que acaba en 0 se vuelva a despertar.

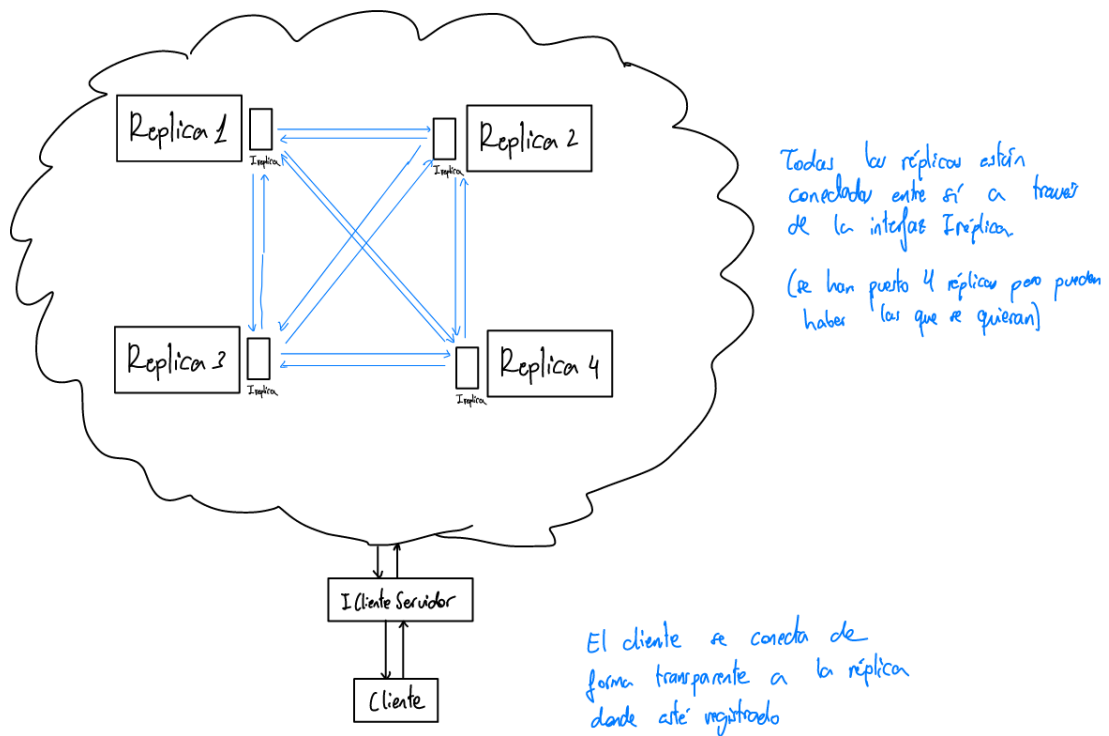
<pre>Ejemplo bound Entra Hebra Cliente 0 Empezamos a dormir Terminamos de dormir Sale hebra Cliente 0 Entra Hebra Cliente 0 Empezamos a dormir Entra Hebra Cliente 1 Sale hebra Cliente 1 Terminamos de dormir Sale hebra Cliente 0</pre>	<pre>Entra Hebra Cliente 8 Sale hebra Cliente 8 Entra Hebra Cliente 10 Empezamos a dormir Terminamos de dormir Sale hebra Cliente 10 Entra Hebra Cliente 2 Sale hebra Cliente 2 Entra Hebra Cliente 13 Sale hebra Cliente 13</pre>
Sin poner <code>synchronized</code>	Poniendo <code>synchronized</code>

- **Ejemplo 3:** Nada especial que comentar. Cuando se ejecuta el cliente aumenta el contador 1000 veces y calcula el tiempo de respuesta.

### Servidor de donaciones

Vamos a montar un servidor con varias réplicas que permitan a los clientes donar dinero a una causa humanitaria.

Hemos establecido 2 interfaces, una para la comunicación entre réplicas (Interfaz `Ireplica`) y entre las réplicas y el cliente (Interfaz `IClienteServidor`). La clase `Replica` implementará estas dos interfaces de manera que se pueda comunicar con el resto de réplicas y con el cliente.



## Funcionamiento

El funcionamiento del servidor desde el punto de vista del cliente se basa en registrarse, iniciar sesión y consultar el total de donaciones. El cliente no sabe a qué réplica está conectado en el momento de hacer cualquier operación.

Cuando se inicia el cliente se le pide que elija una réplica inicial, aunque esta no tiene por qué ser con la que trabaje ya que si está registrado en otra réplica o no está registrado y hay una réplica con menos clientes entonces ya estará trabajando con otra réplica distinta de la que eligió (aunque el cliente no lo sabe).

Todas las réplicas tienen la misma funcionalidad y no existe ninguna superior a las demás. Las réplicas al construirse comprueban si existe un registro en el puerto 1099. Si no existe, lo crea y esa réplica sabe que fue la primera en crearlo. El resto de réplicas al iniciarse, verán que ya existe el registro y lo que harán será conectarse a las réplicas ya creadas.

Todas las réplicas tienen una lista del resto de réplicas de la red que usarán para comprobar donde se debe registrar el usuario o donde debe iniciar sesión.

El mecanismo de agregar réplicas funciona de manera que los datos nuevos se propagan por toda la red de réplicas. De manera que si tenemos 2 réplicas que se tienen en la lista de réplicas mutuamente y agregamos una tercera, esta basta con que se añada cualquiera de las 2 anteriores para que toda la red conozca la nueva réplica. La siguiente imagen ilustra como se propaga el ejemplo comentado.

Situación inicial

replica 1 [replica 2]

replica 2 [replica 1]

Lanzamos "replica 3"

replica 3 [ ]  $\Rightarrow$  replica 3. add Replica (replica 1)

Se añade replica 1 a replica 3 y se propaga.

replica 3 [replica 1]

replica 1. add Replica (replica 3)

replica 1 [replica 2, replica 3]

replica 1 ya tiene la nueva replica

replica 2. add Replica (replica 3)

replica 2 [replica 1, replica 3]

replica 2 ya tiene la nueva replica

replica 3. add Replica (replica 2)

replica 3 [replica 1, replica 2]

replica 3 ya tiene las replicas anteriores

Este mecanismo se usó al principio de la práctica donde al lanzar una réplica hacía falta indicar alguna de las réplicas anteriores para que se agregasen. En la versión final las réplicas agregan todas las réplicas de la red de una vez y de manera recíproca. La propagación no se realiza de forma tan compleja porque se comprueba que ya han sido añadidas y no se vuelven a insertar.

El sistema de localizar el registro y de la autopropagación es muy cómodo porque nos permite lanzar tantas réplicas como y cuando queramos, sin tener que preocuparnos por nada porque se conectan de forma automática. De esta manera podemos tener 3 réplicas iniciales y después de algunas operaciones insertar una cuarta para distribuir la carga de los clientes.

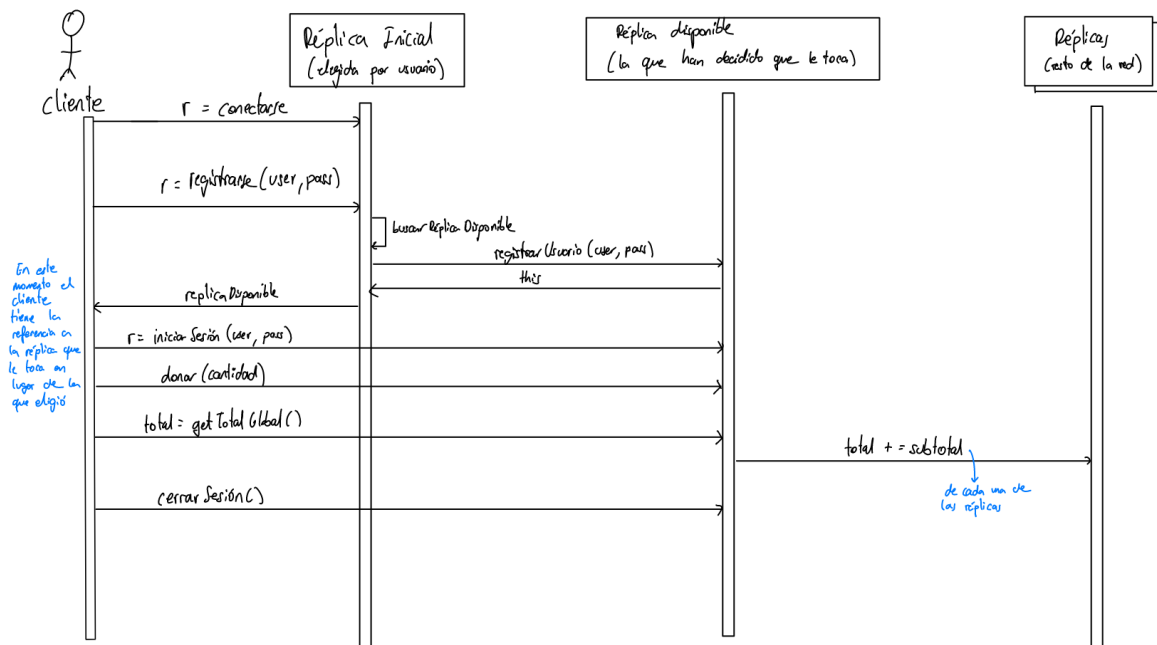
Se han añadido dos funciones a las réplicas: inicio y cierre de sesión. De esta forma el cliente podrá registrar varios usuarios e iniciar sesión dentro de la misma ejecución del programa.

Las réplicas muestran por pantalla (la suya propia, oculta al cliente) mensajes de información sobre la propagación de réplicas o alguna acción del cliente (donación, registro o inicio de sesión).

El curso de una ejecución normal sería el siguiente:

1. El cliente elige una réplica de las disponibles a conectarse
2. El cliente se registra si no tiene cuenta
3. La réplica a la que está conectada decide si es ella u otra la que debe registrar al cliente y le redirige a la que le toque.
4. El usuario queda registrado en la réplica que le tocó y ahora está conectado a esa.
5. El usuario inicia sesión.
6. La réplica comprueba de nuevo que el cliente está en la réplica correcta, lo redirige si es necesario y comprueba los datos de inicio de sesión.
7. Si las credenciales eran correctas, el cliente inicia sesión y ahora tiene disponible la opción de donar, consultar el total y cerrar sesión.
8. El cliente dona en la réplica donde está conectado.
9. El cliente ahora puede consultar el total y lo consulta. La réplica solicita el resto de réplicas sus subtotales y devuelve el total global.
10. El cliente cierra sesión y sale del programa.

Diagrama de secuencia del ejemplo anterior:



Para el cálculo del total, la réplica del cliente solicita al resto de réplicas sus subtotales.

## Ejecución del programa

Primero debemos compilar los ficheros

```
javac *.java
```

Para ejecutar el programa, primero debemos lanzar las réplicas (tantas como queramos). Para lanzar una réplica usaremos la clase "BootReplica" que se encarga de crear réplicas:

```
java -cp . -Djava.security.policy=server.policy BootReplica [id]
```

Donde [id] es un campo opcional donde indicar el id de la réplica (número entero).

Podremos lanzar tantas réplicas como queramos con el comando anterior.

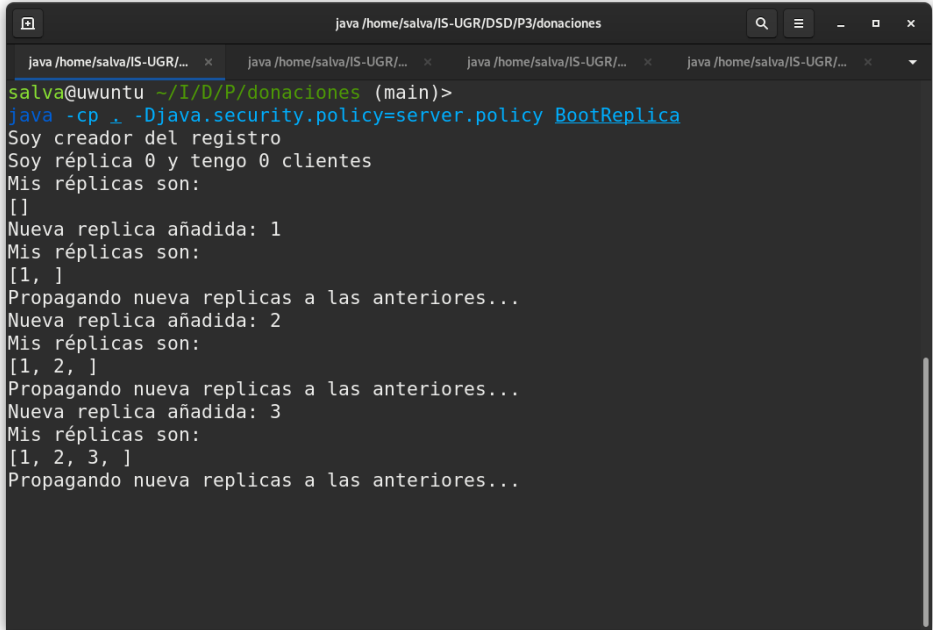
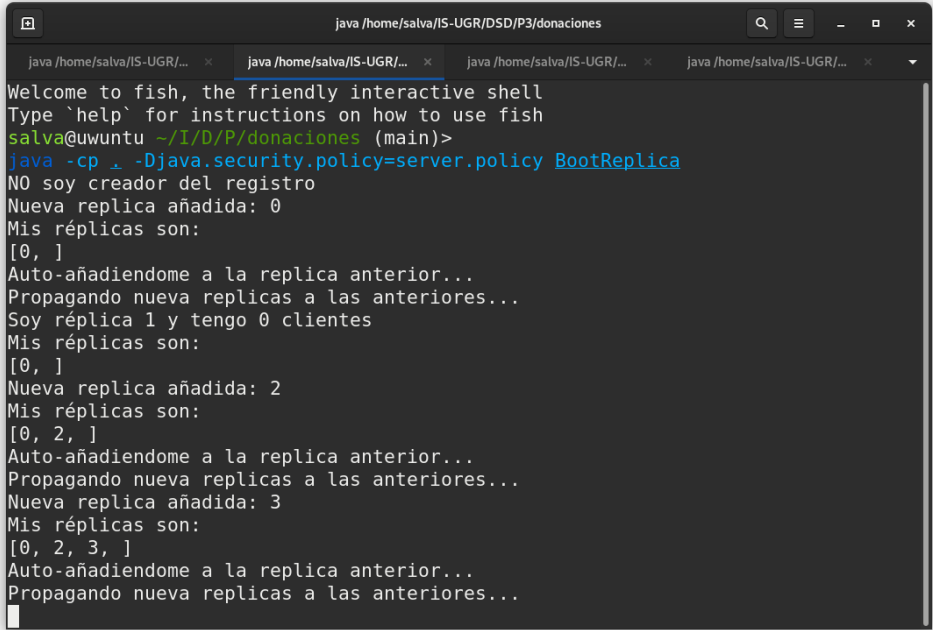
Para lanzar el cliente usaremos

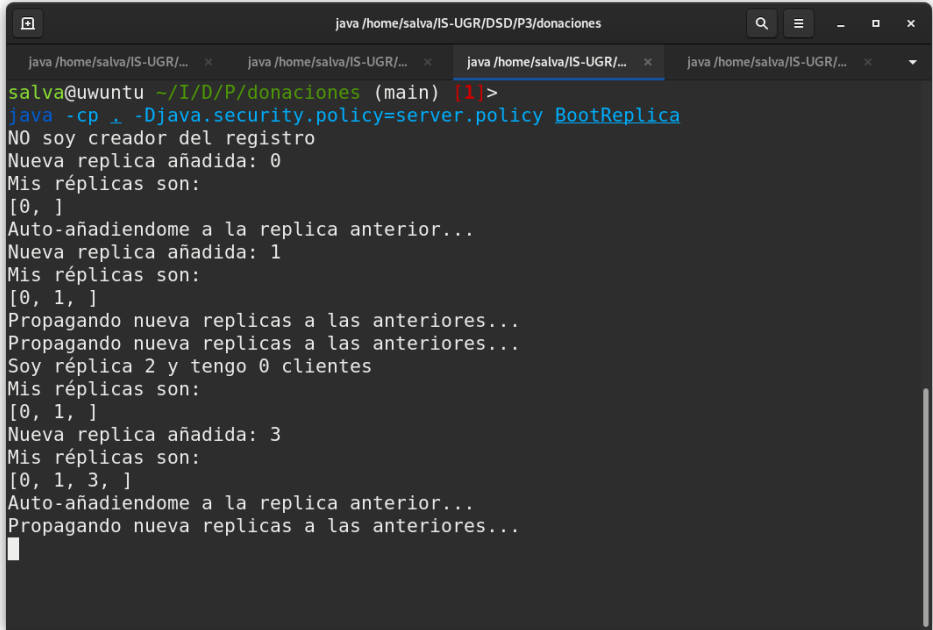
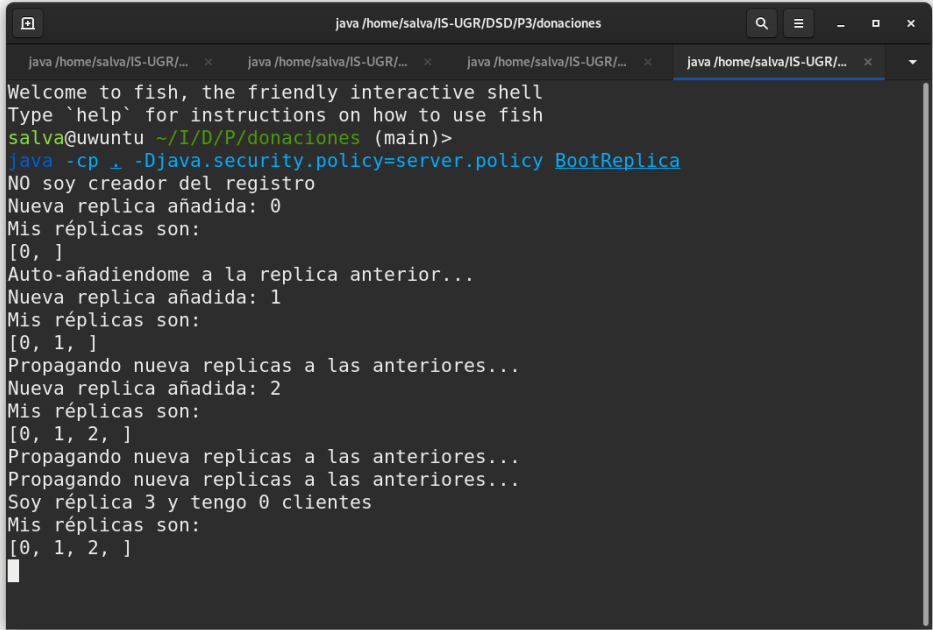
```
java -cp . -Djava.security.policy=server.policy Cliente
```

El programa principal del cliente ya nos guiará para el registro, inicio de sesión y demás operaciones disponibles.

## Capturas

Lanzando réplicas

Réplicas	Captura
Réplica 0	 <pre> java /home/salva/IS-UGR/DSD/P3/donaciones salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica Soy creador del registro Soy réplica 0 y tengo 0 clientes Mis réplicas son: [] Nueva replica añadida: 1 Mis réplicas son: [1, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 2 Mis réplicas son: [1, 2, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 3 Mis réplicas son: [1, 2, 3, ] Propagando nueva replicas a las anteriores... </pre>
Réplica 1	 <pre> java /home/salva/IS-UGR/DSD/P3/donaciones Welcome to fish, the friendly interactive shell Type `help` for instructions on how to use fish salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica NO soy creador del registro Nueva replica añadida: 0 Mis réplicas son: [0, ] Auto-añadiendome a la replica anterior... Propagando nueva replicas a las anteriores... Soy réplica 1 y tengo 0 clientes Mis réplicas son: [0, ] Nueva replica añadida: 2 Mis réplicas son: [0, 2, ] Auto-añadiendome a la replica anterior... Propagando nueva replicas a las anteriores... Nueva replica añadida: 3 Mis réplicas son: [0, 2, 3, ] Auto-añadiendome a la replica anterior... Propagando nueva replicas a las anteriores... </pre>

Réplicas	Captura
Réplica 2	 <pre> java /home/salva/IS-UGR/DSD/P3/donaciones salva@uwuntu ~/I/D/P/donaciones (main) [1]&gt; java -cp _ -Djava.security.policy=server.policy BootReplica NO soy creador del registro Nueva replica añadida: 0 Mis réplicas son: [0, ] Auto-añadiendome a la replica anterior... Nueva replica añadida: 1 Mis réplicas son: [0, 1, ] Propagando nueva replicas a las anteriores... Propagando nueva replicas a las anteriores... Soy réplica 2 y tengo 0 clientes Mis réplicas son: [0, 1, ] Nueva replica añadida: 3 Mis réplicas son: [0, 1, 3, ] Auto-añadiendome a la replica anterior... Propagando nueva replicas a las anteriores... </pre>
Réplica 3	 <pre> Welcome to fish, the friendly interactive shell Type 'help' for instructions on how to use fish salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica NO soy creador del registro Nueva replica añadida: 0 Mis réplicas son: [0, ] Auto-añadiendome a la replica anterior... Nueva replica añadida: 1 Mis réplicas son: [0, 1, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 2 Mis réplicas son: [0, 1, 2, ] Propagando nueva replicas a las anteriores... Propagando nueva replicas a las anteriores... Soy réplica 3 y tengo 0 clientes Mis réplicas son: [0, 1, 2, ] </pre>


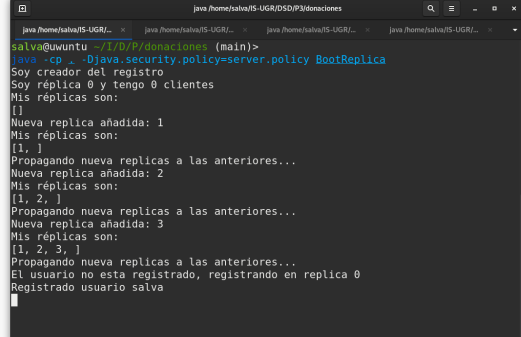
Ejecutando el cliente:

```
java /home/salva/IS-UGR/DSD/P3/donaciones

salva@uwuntu ~/I/D/P/donaciones (main)>
java -cp _ -Djava.security.policy=server.policy Cliente
Introduce la replica a la que quiere conectarse:
0 - replica0
1 - replica1
2 - replica2
3 - replica3

```

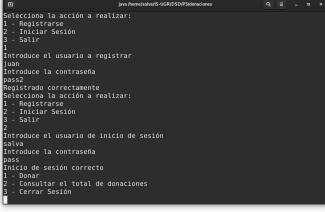
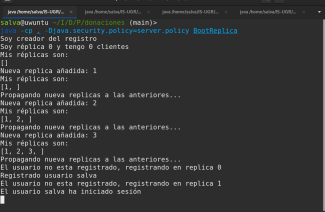
Registrando un usuario:

 <pre>java /home/salva/IS-UGR/DSD/P3/donaciones  java -cp _ -Djava.security.policy=server.policy Cliente Introduce la replica a la que quiere conectarse: 0 - replica0 1 - replica1 2 - replica2 3 - replica3 0 Conectando a replica... Conectado a replica 0 Selecciona la acción a realizar: 1 - Registrarse 2 - Iniciar Sesión 3 - Salir 1 Introduce el usuario a registrar salva Introduce la contraseña pass Registrado correctamente Selecciona la acción a realizar: 1 - Registrarse 2 - Iniciar Sesión 3 - Salir </pre>	 <pre>java /home/salva/IS-UGR/DSD/P3/donaciones  salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica Soy creador del registro Soy réplica 0 y tengo 0 clientes Mis réplicas son: [] Nueva replica añadida: 1 Mis réplicas son: [1, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 2 Mis réplicas son: [1, 2, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 3 Mis réplicas son: [1, 2, 3, ] Propagando nueva replicas a las anteriores... El usuario no está registrado, registrando en replica 0 Registrado usuario salva </pre>
Cliente	Réplica del registro (Réplica 0)

Registrando otro usuario (veremos que se registrará en una réplica distinta)

 <pre>java /home/salva/IS-UGR/DSD/P3/donaciones  1 - Registrarse 2 - Iniciar Sesión 3 - Salir Introduce el usuario a registrar juan Introduce la contraseña paso Registrado correctamente Selecciona la acción a realizar: 1 - Registrarse 2 - Iniciar Sesión 3 - Salir </pre>	 <pre>java /home/salva/IS-UGR/DSD/P3/donaciones  salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica Soy creador del registro Soy réplica 0 y tengo 0 clientes Mis réplicas son: [] Nueva replica añadida: 1 Mis réplicas son: [1, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 2 Mis réplicas son: [1, 2, ] Propagando nueva replicas a las anteriores... Nueva replica añadida: 3 Mis réplicas son: [1, 2, 3, ] Propagando nueva replicas a las anteriores... El usuario no está registrado, registrando en replica 0 Registrado usuario salva El usuario no está registrado, registrando en replica 1 </pre>	 <pre>java /home/salva/IS-UGR/DSD/P3/donaciones  salva@uwuntu ~/I/D/P/donaciones (main)&gt; java -cp _ -Djava.security.policy=server.policy BootReplica Soy creador del registro Soy réplica 1 y tengo 0 clientes Mis réplicas son: [] Auto anadiendome a la replica anterior... Nueva replica añadida: 2 Mis réplicas son: [0, 2, ] Auto anadiendome a la replica anterior... Propagando nueva replicas a las anteriores... Nueva replica añadida: 3 Mis réplicas son: [0, 2, 3, ] Auto anadiendome a la replica anterior... Propagando nueva replicas a las anteriores... Registrado usuario juan </pre>
Cliente	Réplica 0 (réplica del anterior registro)	Réplica 1 (la que se ha decidido que debe hacer el registro)

Si ahora iniciamos sesión con "salva" (se registró en réplica 0) veremos que se volverá a cambiar a la réplica 0.

 <pre>java -Dname=CLIENT -jar jms-Replicas-0.0.0-RC3.jar Selecciona la acción a realizar: 1 - Registrarse 2 - Salir Sesión 3 - Salir 1 Introduce el usuario a registrar: juan Introduce la contraseña: pass Registrado correctamente. Selecciona la acción a realizar: 1 - Registrarse 2 - Iniciar Sesión 3 - Salir 2 Introduce el usuario de inicio de sesión: juan Introduce la contraseña: pass Inicio de sesión correcto 1 - Donar 2 - Consultar el total de donaciones 3 - Cerrar Sesión</pre>	 <pre>java -Dname=REP0 -jar jms-Replicas-0.0.0-RC3.jar Introduce el monto a donar: 10 Introduce el usuario: Juan Introduce la contraseña: pass Nueva replicación creada en esta réplica Nueva réplica 0 y tengo 0 clientes Mis réplicas son: [] Nueva réplica añadida: 1 Mis réplicas son: [1] Propagando nueva réplica a las anteriores... Nueva réplica añadida: 2 Mis réplicas son: [1, 2] Propagando nueva réplica a las anteriores... Nueva réplica añadida: 3 Mis réplicas son: [1, 2, 3] Propagando nueva réplica a las anteriores... El usuario no está registrado, registrando en replica 0 Registrado usuario juan El usuario no está registrado, registrando en replica 1 El usuario juan ha iniciado sesión</pre>	 <pre>java -Dname=REP1 -jar jms-Replicas-0.0.0-RC3.jar Introduce el monto a donar: 20 Introduce el usuario: Juan Introduce la contraseña: pass Nueva replicación creada en esta réplica Nueva réplica 1 y tengo 0 clientes Mis réplicas son: [] Introduciendome a la réplica anterior... Propagando nueva réplica a las anteriores... Nueva réplica añadida: 0 Mis réplicas son: [0, 1] Introduciendome a la réplica anterior... Propagando nueva réplica a las anteriores... Nueva réplica añadida: 2 Mis réplicas son: [0, 2, 1] Introduciendome a la réplica anterior... Propagando nueva réplica a las anteriores... Nueva réplica añadida: 3 Mis réplicas son: [0, 2, 3] Introduciendome a la réplica anterior... Propagando nueva réplica a las anteriores... Registrado usuario juan El usuario no está registrado en esta réplica, redirigiendo a replica 0</pre>
Cliente	Réplica 0 (réplica donde se registró)	Réplica 1 (donde estábamos actualmente)

Ejemplo de donación: ambos donan cantidades distintas (y en réplicas distintas)



```
Introduce el usuario de inicio de sesión
salva
Introduce la contraseña
pass
Inicio de sesión correcto
1 - Donar
2 - Consultar el total de donaciones
3 - Cerrar Sesión
1
Introduce la cantidad a donar
1000
Se han donado 1000.0 a la causa humanitaria
1 - Donar
2 - Consultar el total de donaciones
3 - Cerrar Sesión
3
Selecciona la acción a realizar:
1 - Registrarse
2 - Iniciar Sesión
3 - Salir
2
Introduce el usuario de inicio de sesión
juan
Introduce la contraseña
pass2
Inicio de sesión correcto
1 - Donar
2 - Consultar el total de donaciones
3 - Cerrar Sesión
1
Introduce la cantidad a donar
500
Se han donado 500.0 a la causa humanitaria
1 - Donar
2 - Consultar el total de donaciones
3 - Cerrar Sesión
2
El total de donaciones es 1500.0
```