

Práctica 3. Lenguajes y *Middlewares* para Programación Distribuida: Java RMI

Analizando los ejemplos

Ejemplo 1

El cliente solicita al servidor usar el método remoto **escribir_mensaje**.

Para ello primero mira si se ha establecido algún tipo de seguridad en el cliente. De no ser así se le crea uno. Luego mira el ***rmiregistry*** en busca del objeto **Ejemplo_I** de la clase **Ejemplo_I** para crear una instancia local. Por último se llama al método desde la instancia local de ese objeto.

El método **escribir_mensaje** mira el número del proceso que pasamos como argumento al cliente. Si es 0 hace una espera de 5 segundos. En cualquier caso, imprimirá el número de hebra.

Ejemplo 2

Se pasa por argumento el host y el número de hilos a ejecutar. El cliente lanza los hilos al servidor, y si el número de hilo termina en 0 se le pone un delay de 5 segundos.

Si se pone antes **synchronized**, la ejecución de los hilos es secuencial. Esto es, se lanza el hilo y hasta que no acabe su ejecución no sigue con el siguiente.

Ejemplo 3

En el servidor hay una clase **contador** que tiene implementada la interfaz **icontador**. El servidor sólo crea una instancia de contador y bindea el mensaje.

En el cliente se trata de asignar el número a 0 e incrementarlo 1000 veces, contando el tiempo entre ambas.

Al final imprime en pantalla el tiempo empleado en realizar las operaciones y el número de operaciones.

En general, de los ejemplos podemos sacar:

- Los servidores heredan de una clase con la implementación de una interfaz. Si bien el servidor sólo trata de establecer la conexión, se puede hacer uso de las operaciones implementadas.
- El cliente puede acceder a todos los métodos remotos que se extienden al servidor
- Usar **synchronized** permite la ejecución secuencial de los métodos.

Algo interesante es que el registro debe reiniciarse a la hora de ejecutar cada ejemplo. Desconozco el por qué.

El servidor de donaciones

El servidor de donaciones se basa en la actualización constante de la información que hay en los servidores.

La interfaz de donaciones

La interfaz de donaciones se ocupa de definir las operaciones que luego se tendrán en cuenta para su posterior implementación.

Nótese que ya que

Operaciones básicas

El servidor que luego extiende los programas servidor consta de varios atributos:

- Un conjunto de entidades de cada servidor, con su nombre y cantidad total donada.
- Una serie de registros de donaciones
- La IP del host del server
- El nombre del servidor actual.
- Una serie de Réplicas, que contienen el nombre y la copia del servidor. Si bien al principio sólo contiene los nombres, con ciertos métodos se podrá hacer la asociación.

Dicho esto, listo alguno una serie de operaciones de soporte a los básicos:

- ✓ **getNombre** : Dice el nombre del servidor.
- ✓ **comprobarRegistroEntidad** : Comprueba si la entidad existe en cualquiera de los servidores
- ✓ **buscarEntidad** : Busca entre todos los servidores la entidad y retorna la copia de la entidad.
- ✓ **getNumEntidades** : Recoge el número de entidades que hay en un servidor
- ✓ **isBloqueado** : Mira si el servidor ha realizado ya una operación. Sirve de candado para evitar la replicación de la operación a las referencias de este servidor en las réplicas.
- ✓ **setBloqueado** : Cambia el estado del candado explicado en el método anterior.
- ✓ **unlockAll** : Desbloquea todos los servidores. Se recomienda usar después de realizar las operaciones (normalmente después de llamar al método)
- ✓ **getReplica** : Dada una réplica con el nombre, se oibtiene la referencia al servidor y se asigna a la Réplica.
- ✓ **hayRegistros** : Comprueba si hay registros en el servidor
- ✓ **modificarEntidad** : Como cogemos una copia de la entidad, se asigna al ArrayList de Entidades.

Y con ello, la lista de operaciones básicas:

- ✓ **registrarEntidad** : Crea una Entidad en el servidor con menos Entidades registradas. Si tienen la misma cantidad, se registraría en el servidor desde donde se conecta.
- ✓ **iniciarSesionEntidad** : Con una Entidad registrada (se comprueba), se trata de iniciar sesión. Para ello sólo hace falta poner el usuario (aunque tenía pensado

poner contraseña, sólo haría falta introducir en el constructor la contraseña y se comprobaría en este mismo método)

- ✓ **realizarDonacion** : Se realiza una donación. Para evitar la corrupción de las entidades (y evitar que cojan dinero de la causa), la cantidad deberá ser mayor de 0 para poder crear un registro.
- ✓ **consultarTotal** : Mira la entidad para ver el total donado en todos los servidores
- ✓ **consultarSubtotal** : Mira los registros del servidor para hacer la sumatoria y ver cuánto se ha donado en ese servidor.

Operaciones extra

Como extra, el método **listarRegistros** permite recoger todos los movimientos de la entidad en el servidor concreto.

Los programas servidor

Los programas servidores siguen el mismo esquema, pero las réplicas cambian una línea.

El servidor primero crea el registro y está a la escucha de los eventos del Servidor de Donaciones. Además se hace un *bind* del servidor al registro.

En el caso de las réplicas, cambian los servidores que, según la réplica, serían sus réplicas; y que no tendría la línea para crear el Registro, pues ya estaría creado (sólo habría que *bindearlo*)

```
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ java -cp . -Djava.security.policy=server.policy Cliente
Por favor, especificame a qué servidor quieres conectarte poniendo el nombre de éste después del nombre de la clase.
Estos son los servidores a los que puedes conectarte:
server1
server2
server3
```

El programa cliente

El programa cliente es capaz de poder acceder a las funcionalidades principales.

Para acceder, se hace uso de **java -cp . -**

Djava.security.policy=server.policy Cliente <servidor> . Si no se sabe qué servidores son, si no se pone el último argumento, te pondrá el listado de servidores.

Si no se ha iniciado sesión en la entidad, te permitirá registrar e iniciar sesión de una entidad. Además se puede salir del programa.

En caso de iniciar sesión, se podrá donar a la causa, consultar subtotales y totales, consultar los movimientos en el servidor, o salir del programa.

Pruebas y errores

Uno de los errores que se dio al principio del desarrollo es una excepción por Serialización. Pero fue cuestión de implementar la interfaz Serializable.

Otro error que he podido apreciar, que pasa aleatoriamente, es un error por acceso concurrente. Sin embargo, no he sido capaz de replicar el error, por lo que no sabría cómo poder arreglarlo. No debería ser mayor problema siempre y cuando se haya ejecutado el comando **rmiregistry 1099**.

1. Ejecución y registro de varias entidades.

```
Cliente
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ java -cp . -Djava.security.policy=server.policy Cliente
Por favor, especificame a qué servidor quieres conectarte poniendo el nombre de éste después del nombre de la clase.
Estos son los servidores a los que puedes conectarte:
server1
server2
server3
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ java -cp . -Djava.security.policy=server.policy Cliente server1
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir

R
Dime el nombre de la entidad
Entidad A
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir

R
Dime el nombre de la entidad
Entidad B
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir

R
Dime el nombre de la entidad
Entidad C
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir
```

```
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server1
>> Server1 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad A Total donado : 0.0€.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
```

```
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server2
>> Server2 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad B Total donado : 0.0€.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
```

```
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server3
>> Server3 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad C Total donado : 0.0€.
```

2. Donar desde servidor donde estaba la Entidad registrada, y uso de las funciones

```

Cliente
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ java -cp . -Djava.security.policy=server.policy Cliente server1
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir

L
Dime el nombre de la entidad
Entidad A
Se ha logueado la entidad.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

D
¿Cuánto desea donar a la causa?
10000
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

B
La entidad Entidad A ha donado 10000.0€ a la causa en este servidor.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

T
La entidad Entidad A ha donado 10000.0€ a la causa.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

M
Host      - Nombre servidor  - Cantidad
-----
localhost server1 10000.0

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server1
>> Server1 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad A Total donado : 0.0€.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
Iniciando sesión con Entidad A
Entidad A ha donado 10000.0€ a la causa
[Entidad] Nombre : Entidad A Total donado : 10000.0€.
[Registro] Entidad A , Dinero donado : 10000.0
Registro hallado

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server2
>> Server2 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad B Total donado : 0.0€.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server3
>> Server3 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad C Total donado : 0.0€.

```

3. Donar desde servidor donde NO estaba la Entidad registrada, y uso de las funciones

```

Cliente
ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ java -cp . -Djava.security.policy=server.policy Cliente server3
Conectado desde -> server3
Bienvenido al sistema. Elige una opción
    R -> Registrar una entidad
    L -> Iniciar sesión de una entidad registrada
    S -> Salir

L
Dime el nombre de la entidad
Entidad A
Se ha logueado la entidad.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

D
¿Cuánto desea donar a la causa?
2000
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

B
La entidad Entidad A ha donado 2000.0€ a la causa en este servidor.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

T
La entidad Entidad A ha donado 12000.0€ a la causa.
Logueado como Entidad A. Elige una opción
    D -> Donar a la causa
    B -> Mirar cuánto se ha donado en este servidor
    T -> Mirar cuánto se ha donado en total
    M -> Mirar el historial de donaciones en este servidor
    S -> Salir

M
Host      - Nombre servidor - Cantidad
-----
localhost server3 2000.0

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server1
>> Server1 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad A Total donado : 0.0€.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
Iniciando sesión con Entidad A
Entidad A ha donado 10000.0€ a la causa
[Entidad] Nombre : Entidad A Total donado : 10000.0€.
[Registro] Entidad Entidad A , Dinero donado : 10000.0
Registro hallado

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server2
>> Server2 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad B Total donado : 0.0€.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.

```

```

ivan@Arcadia-TUF:/mnt/c/Users/Ivan/Desktop/Clase/DSD/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server3
>> Server3 listo
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad B en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
[!] No he encontrado la entidad Entidad C en mi server.
Se ha creado la entidad -> [Entidad] Nombre : Entidad C Total donado : 0.0€.
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
Iniciando sesión con Entidad A
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
Entidad A ha donado 2000.0€ a la causa
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.
[Entidad] Nombre : Entidad A Total donado : 12000.0€.
[Registro] Entidad Entidad A , Dinero donado : 2000.0
Registro hallado
[!] No he encontrado la entidad Entidad A en mi server.
[!] No he encontrado la entidad Entidad A en mi server.

```

4. Otras pruebas: Inicio de sesión con Entidad inexistente, Donaciones negativas...

```

Cliente
iente server1
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
  R -> Registrar una entidad
  L -> Iniciar sesión de una entidad registrada
  S -> Salir

R
Dime el nombre de la entidad
A
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
  R -> Registrar una entidad
  L -> Iniciar sesión de una entidad registrada
  S -> Salir

R
Dime el nombre de la entidad
B
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
  R -> Registrar una entidad
  L -> Iniciar sesión de una entidad registrada
  S -> Salir

R
Dime el nombre de la entidad
C
Se ha registrado la entidad.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
  R -> Registrar una entidad
  L -> Iniciar sesión de una entidad registrada
  S -> Salir

L
Dime el nombre de la entidad
D
La entidad no existe.

Cliente
La entidad no existe.
Conectado desde -> server1
Bienvenido al sistema. Elige una opción
  R -> Registrar una entidad
  L -> Iniciar sesión de una entidad registrada
  S -> Salir

L
Dime el nombre de la entidad
A
Se ha logueado la entidad.
Logueado como A. Elige una opción
  D -> Donar a la causa
  B -> Mirar cuánto se ha donado en este servidor
  T -> Mirar cuánto se ha donado en total
  M -> Mirar el historial de donaciones en este servidor
  S -> Salir

D
¿Cuánto desea donar a la causa?
-1234
Logueado como A. Elige una opción
  D -> Donar a la causa
  B -> Mirar cuánto se ha donado en este servidor
  T -> Mirar cuánto se ha donado en total
  M -> Mirar el historial de donaciones en este servidor
  S -> Salir

Opción inválida
Logueado como A. Elige una opción
  D -> Donar a la causa
  B -> Mirar cuánto se ha donado en este servidor
  T -> Mirar cuánto se ha donado en total
  M -> Mirar el historial de donaciones en este servidor
  S -> Salir

```

```

ivan@Arcadia-TUF: /mnt/c/Users/Ivan/Desktop/Clase/DSO/Practica/P3/Ejercicio/build$ k
ll 257
ivan@Arcadia-TUF: /mnt/c/Users/Ivan/Desktop/Clase/DSO/Practica/P3/Ejercicio/build$ r
iregistry 1099 &
[1] 413
ivan@Arcadia-TUF: /mnt/c/Users/Ivan/Desktop/Clase/DSO/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server1
>> Server1 listo
[!] No he encontrado la entidad A en mi server.
[!] No he encontrado la entidad A en mi server.
Se ha creado la entidad -> [Entidad] Nombre : A Total donado : 0.0€.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad D en mi server.
[!] No he encontrado la entidad D en mi server.
Iniciando sesión con A
[!] Han intentado sacar dinero de una causa benéfica >:(

```

```

ivan@Arcadia-TUF: /mnt/c/Users/Ivan/Desktop/Clase/DSO/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server2
>> Server2 listo
[!] No he encontrado la entidad A en mi server.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad B en mi server.
Se ha creado la entidad -> [Entidad] Nombre : B Total donado : 0.0€.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad D en mi server.

```

```

ivan@Arcadia-TUF: /mnt/c/Users/Ivan/Desktop/Clase/DSO/Practica/P3/Ejercicio/build$ j
ava -cp . -Djava.security.policy=server.policy Server3
>> Server3 listo
[!] No he encontrado la entidad A en mi server.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad B en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad C en mi server.
[!] No he encontrado la entidad D en mi server.
Se ha creado la entidad -> [Entidad] Nombre : C Total donado : 0.0€.
[!] No he encontrado la entidad D en mi server.

```

¿Y cómo se prueba?

Para compilar los archivos se ha creado el script BuildIt.sh, que mueve los objetos (*.class) a la carpeta **build**, y copia el archivo de políticas a esa carpeta.

Para ejecutar los archivos, se haría uso de:

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

Para ejecutar los servidores, y en el caso del Cliente:

```
java -cp . -Djava.security.policy=server.policy Cliente <servidor>
```

donde si se quita la zona en brackets se puede ver los servers, y una vez puesto el server de la lista, se ejecuta el menú.