

Proyecto RMI

Servicio de alertas centralizado



Sistemas Distribuidos y Servicios Web

Ana Lucero Fernández
Carlos Rodríguez Hernández

Índice

1.- Motivación.....	3
2.- Introducción.....	3
3.- Implementación.....	3
3.1.- Clientes.....	3
3.2.- Centralita.....	4
3.3.- Observadores.....	4
4.- Compilación y ejecución.....	5
5.- Funcionamiento.....	6
6.- Relación con la asignatura.....	7
7.- Referencias.....	7

1.- Motivación

En este primer trabajo de la asignatura se pide realizar un proyecto basado en RMI usando los conceptos de sistemas distribuidos que se han dado a lo largo de las primeras 6 prácticas. Por tanto, a lo largo de este trabajo se han usado todos los métodos y técnicas que se han visto en estas primeras prácticas. En esta memoria se incluye una explicación del trabajo realizado, así como su funcionamiento, en las referencias se puede obtener información detallada sobre los conceptos teóricos vistos en la asignatura sobre RMI.

2.- Introducción

Se pretende diseñar un sistema de alertas de emergencias centralizado (estilo 112) en el que se disponga de un servicio de centralita al que llegan los diferentes mensajes de los clientes. A esta centralita están conectados, como observadores, los diferentes Servicios de Emergencia (bomberos, sanitarios, guardia civil, policía local, policía nacional, etc), de manera que cuando la centralita recibe un mensaje con alguna incidencia, determina a qué Servicio o Servicios redirigir esta incidencia. En paralelo, la centralita incorpora un servicio de log para dejar constancia de todos los eventos procesados.

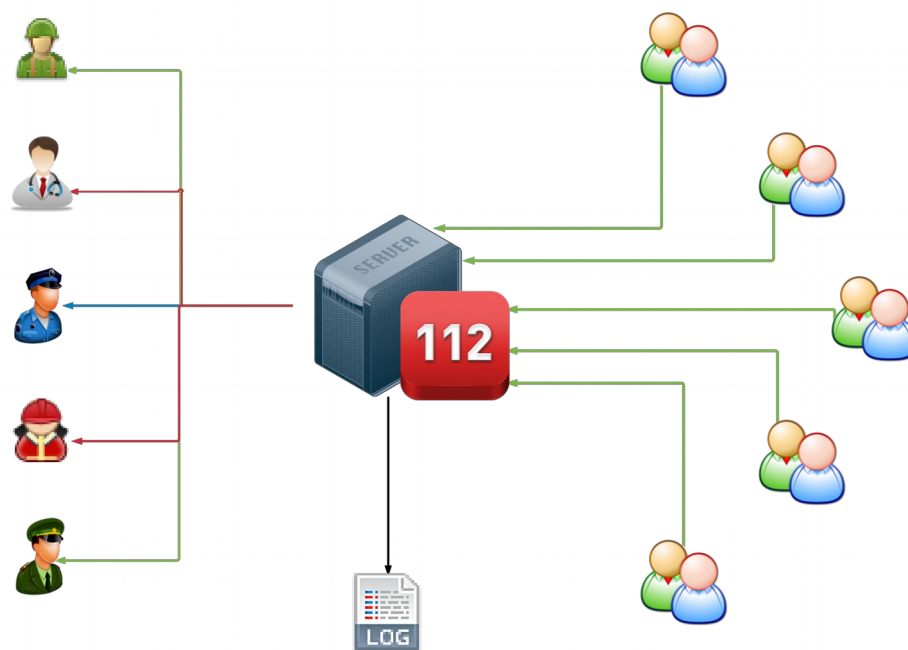


Ilustración 1: Esquema de funcionamiento general

3.- Implementación

Este proyecto consta de 3 partes diferenciadas:

3.1.- Clientes

Los clientes son la parte más numerosa en cuanto a cantidad de dispositivos, puesto que simulan a los usuarios habituales en la vida real que podría tener el servicio, por tanto puede ser una cifra muy elevada la que tenga acceso de manera simultánea.

Su funcionamiento consiste en mandar un mensaje de texto a la centralita en el que se dan detalles de la situación que requiere la presencia de los Servicios de Emergencia, así como alguna información extra como la dirección del suceso, un número de teléfono, etc.

En la ejecución del programa el usuario mantendrá una conversación guiada en tiempo real con la centralita, la cual va registrando las respuestas dadas por el usuario. Una vez que finalice la entrada de texto y envíe el mensaje, esperará hasta recibir confirmación sobre los Servicios de Emergencia avisados, finalizando así su ejecución.

3.2.- Centralita

La centralita es un único elemento que hace la función de servidor, controlador, distribuidor y registro de log.

Su funcionamiento es el siguiente:

1. Recibe un mensaje por parte de algún cliente.
2. Extrae los diferentes campos de ese mensaje (nombre, teléfono, dirección, mensaje, etc).
3. Mediante expresiones regulares trata el campo "mensaje" para determinar a qué Servicio o Servicios de Emergencia redirigir la alerta. Por ejemplo, si detecta que en el campo texto se hace uso de las palabras *fuego*, *incendio*, *humo*, etc; la centralita determinará que esa alerta debe dirigirla hacia los bomberos.
4. Una vez que sabe a qué Servicio de Emergencia avisar, forma el mensaje de salida y lo manda hacia el Servicio en cuestión. De igual manera avisa al usuario con los Servicios que han sido alertados.

Aparte de avisar a los diferentes Servicios de Emergencia, la centralita debe llevar el control de cuáles Servicios están registrados como observadores, registrando así cualquier alta o baja.

Por último, la centralita también realiza un control interno tanto de los mensajes recibidos como de los mensajes enviados haciendo uso de servicios de log y mostrando por pantalla toda esta información.

3.3.- Observadores

Los observadores son los diferentes Servicios de Emergencia que están suscritos a las notificaciones de la centralita, en este caso está restringido a 4 actores (Policía, Bomberos, Sanitarios y Guardia Civil), pudiéndose registrar todos ellos a la vez, incluso varios observadores del mismo tipo simultáneamente.

El funcionamiento de los observadores se inicia con la petición de añadirse a la lista de la centralita, una vez adscritos esperan la recepción de un mensaje con los datos sobre alguna alerta. Este mensaje se muestra por pantalla para que los profesionales del Servicio en cuestión que estarían tras el ordenador puedan actuar y acudir al lugar del suceso.

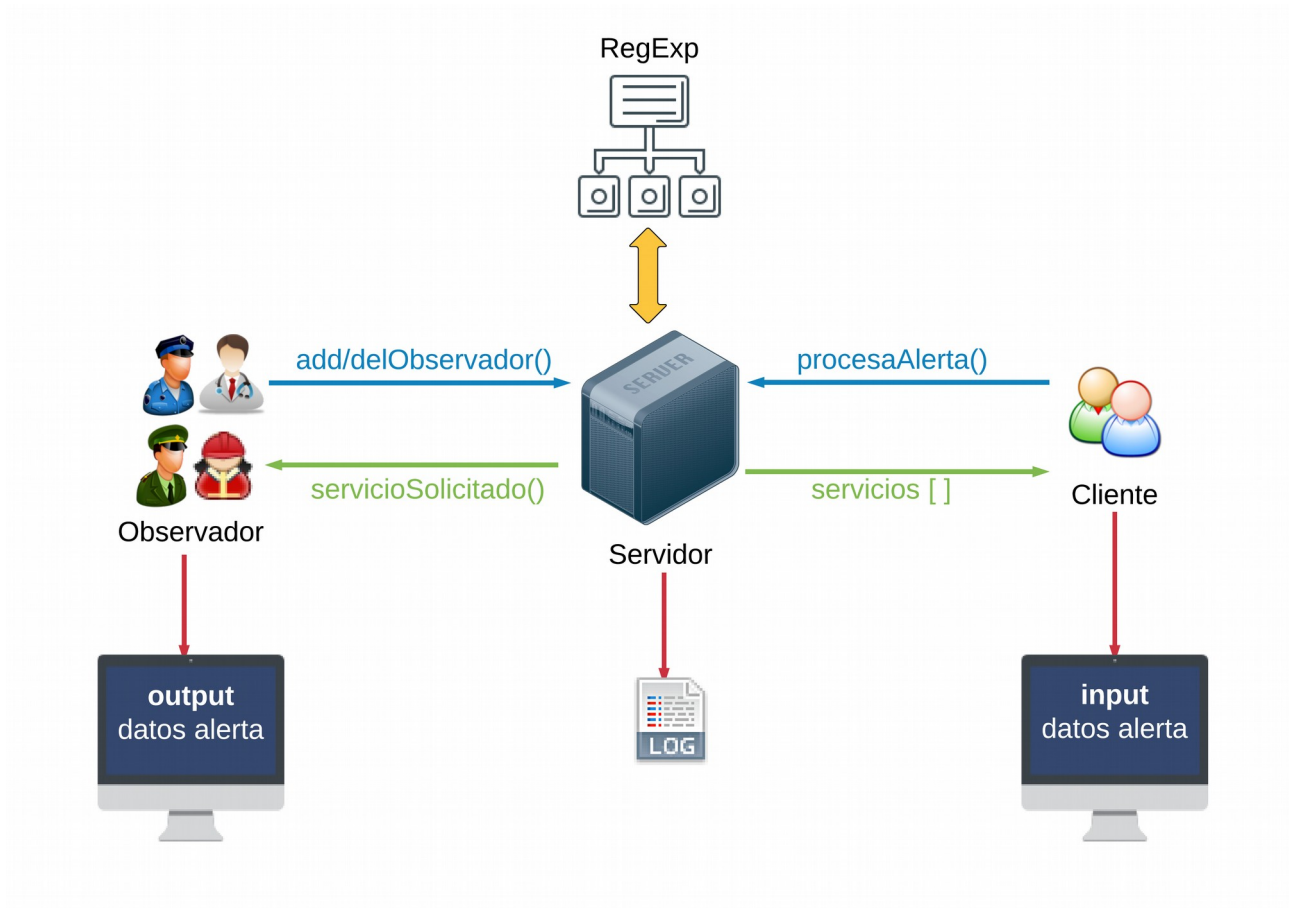


Ilustración 2: Esquema de comunicación

4.- Compilación y ejecución

Se ha desarrollado un script para realizar una compilación limpia y automática, los diferentes makefiles copian los ficheros .class necesarios de un directorio a otro en el momento adecuado:

```
bash ./compila.sh
```

Para ejecutar este proyecto hay que hacerlo en diferentes terminales.

Terminal 1 (centralita):

```
cd centralita  
rm registry 54321 &  
make exeCentralita
```

Terminal 2 (observador, se pueden levantar tantos como se quiera):

```
cd observador  
make exeObservador
```

Terminal 3 (cliente, se pueden levantar tantos como se quiera):

```
cd cliente  
make exeCliente
```

NOTA: Para ejecutarlo en diferentes equipos hay que cambiar en los *makefiles* la IP en la orden de ejecución. Sustituir *localhost* por las IPs deseadas.

5.- Funcionamiento

Ejecución del cliente por parte de un usuario. Se puede observar las preguntas de la centralita y respuestas del usuario, siendo una conversación fluida, automatizada y en tiempo real. Como se puede ver, el mensaje hace referencia a un incendio con heridos, por tanto la centralita determina que hay que avisar a Bomberos y Sanitarios:

```
carlos@lenovo /home/carlos/Dev/US/SDSW/proyectoRMI/cliente <system> <master*>
$ make exeCliente
java -Djava.security.policy=cliente.permisos ClienteCentralita localhost 54321

Centralita> Bienvenido al servicio de alerta ¿Cuál es su nombre? Carlos

Centralita> ¿Podría indicar un teléfono de contacto?
Carlos> 6321543258

Centralita> ¿Cuál es la dirección del suceso?
Carlos> Calle San Jacinto (Sevilla)

Centralita> Detalle a continuación la situación que requiere nuestra intervención:
Carlos> Hay un incendio con mucho humo y varias personas heridas en la calle

Centralita> Se ha avisado a los siguientes servicios:
                [1] Bomberos
                [2] Sanitarios
```

Ilustración 3: Ejecución cliente

Registro de 4 observadores (cada uno identificado por su nombre y código de color):

1	2	3	4
<pre>make exeObservador java -Djava.security.policy=observador.permisos Observador Centralita localhost 54321 Elija el servicio: [1] Bomberos [2] Sanitarios [3] Policia [4] Guardia Civil 1 Observador Bomberos registrado en la centralita Para salir, pulsar Ctrl+D</pre>	<pre>make exeObservador java -Djava.security.policy=observador.permisos Observador Centralita localhost 54321 Elija el servicio: [1] Bomberos [2] Sanitarios [3] Policia [4] Guardia Civil 2 Observador Sanitarios registrado en la centralita Para salir, pulsar Ctrl+D</pre>	<pre>make exeObservador java -Djava.security.policy=observador.permisos Observador Centralita localhost 54321 Elija el servicio: [1] Bomberos [2] Sanitarios [3] Policia [4] Guardia Civil 3 Observador Policia registrado en la centralita Para salir, pulsar Ctrl+D</pre>	<pre>make exeObservador java -Djava.security.policy=observador.permisos Observador Centralita localhost 54321 Elija el servicio: [1] Bomberos [2] Sanitarios [3] Policia [4] Guardia Civil 4 Observador Guardia Civil registrado en la centralita Para salir, pulsar Ctrl+D</pre>

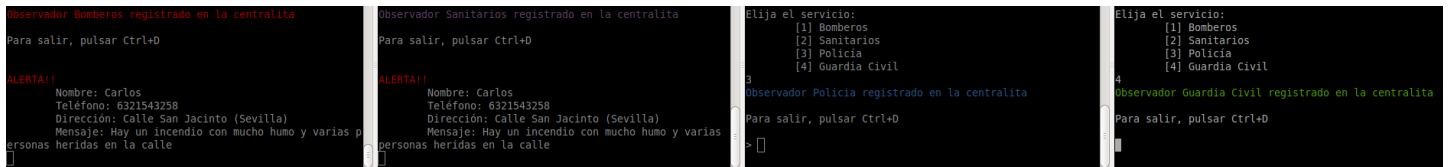
Ilustración 4: Ejecución observadores

Ejecución de la centralita en la que se ve el registro/baja de los observadores y cuando se hace match con algunas palabras clave, lo cual indica que hay que avisar a esos Servicios:

```
carlos@lenovo /home/carlos/Dev/US/SDSW/proyectoRMI/centralita <system> <master*>
$ make exeCentralita
java -Djava.security.policy=servidor.permisos Servidor 54321
Añadido Observador Bomberos
Añadido observador Sanitarios
Añadido observador Policia
Añadido observador Guardia Civil
Match con Bomberos
Match con Sanitarios
Eliminado observador Policia
```

Ilustración 5: Ejecución centralita

Alerta en los observadores avisados (Bomberos y Sanitarios) y no en el resto:



```
Observador Bomberos registrado en la centralita
Para salir, pulsar Ctrl+D

ALERTA!!
Nombre: Carlos
Teléfono: 6321543258
Dirección: Calle San Jacinto (Sevilla)
Mensaje: Hay un incendio con mucho humo y varias personas heridas en la calle

Observador Sanitarios registrado en la centralita
Para salir, pulsar Ctrl+D

ALERTA!!
Nombre: Carlos
Teléfono: 6321543258
Dirección: Calle San Jacinto (Sevilla)
Mensaje: Hay un incendio con mucho humo y varias personas heridas en la calle

Elija el servicio:
[1] Bomberos
[2] Sanitarios
[3] Policía
[4] Guardia Civil

Observador Policía registrado en la centralita
Para salir, pulsar Ctrl+D
>

Elija el servicio:
[1] Bomberos
[2] Sanitarios
[3] Policía
[4] Guardia Civil

Observador Guardia Civil registrado en la centralita
Para salir, pulsar Ctrl+D
>
```

Ilustración 6: Ejecución observadores recibiendo alerta

6.- Relación con la asignatura

Como se puede observar, en este proyecto se va a hacer uso de casi todos los elementos vistos en las prácticas, utilizando conceptos de todas ellas, tales como aspectos relacionados con el servicio de chat, la fábrica de logs, el servicio de alarmas, las marcas de tiempo, etc.

7.- Referencias

Debido a que este trabajo se ha desarrollado siguiendo la base teórica vista en las prácticas de esta asignatura, se ha evitado explicar en esta memoria los conceptos teóricos relativos a RMI, para ello se incluyen las referencias a los enunciados de la asignatura.

<https://github.com/carrodher/SDSW/blob/master/practica1/P1RPC.pdf>

<https://github.com/carrodher/SDSW/blob/master/practica2/P2RPCfs.pdf>

<https://github.com/carrodher/SDSW/blob/master/practica3/P3RMI.pdf>

<https://github.com/carrodher/SDSW/blob/master/practica4/P4RMIRreferencias.pdf>

<https://github.com/carrodher/SDSW/blob/master/practica5/P5RMIRreferenciasII.pdf>

<https://github.com/carrodher/SDSW/blob/master/practica6/P6RMIOobjetos.pdf>