



Objetivo de la práctica

El objetivo de la práctica consiste en implementar una aplicación distribuida para el envío de mensajes entre usuarios del estilo **WhatsApp**. La aplicación se basará en el paradigma de objetos distribuidos y utilizando Java RMI.



Enunciado de la práctica

La aplicación de mensajes utilizará una interface basado en texto. La aplicación implementará un sistema orientado de mensajes (MOM) para enviar y recibir los mensajes, utilizando los callbacks de RMI.

El cliente de la aplicación solo tendrá un parámetro opcional, mediante el cual se podrá definir la IP/Url del servidor en donde se ejecutan sus objetos remotos.

Sintaxis: MsgRMI [<ip_servidor>]

La aplicación soportará la siguiente funcionalidad:

- Login: Se permitirá conectarse a la aplicación mediante un usuario y contraseña. La aplicación pedirá la contraseña y la validará contra el servidor. Cada vez que se conecta un nuevo usuario al sistema, este envía un mensaje informativo a todos los usuarios conectados notificando este hecho.

Sintaxis: Login <Usuario> [<Contraseña>]

- Logout: Desconecta el usuario del sistema, pero no cierra la aplicación de mensajes.

Sintaxis: Logout

- Crear usuario: Permite crear un nuevo usuario en el sistema, especificando su nombre y contraseña y opcionalmente el grupo al cual pertenece.

Sintaxis: NewUser <Usuario> <Contraseña> [<Grupo>]

- Enviar mensaje: Permitirá enviar un mensaje tanto a un usuario concreto, como a un grupo de usuarios.

Sintaxis: SendMsg <Usuario> [-g <Grupo>] <TextoMensaje>

- Recibir mensaje: La aplicación mostrará los mensajes enviados al usuario, indicando el usuario que lo envía, la hora y el texto del mensaje.

- Crear grupo de usuarios: Permite crear un grupo de usuarios a los cuales se pueden unir los usuarios para recibir los mensajes enviados al grupo (modelo subscripción/publicación).

Cada vez que se crea un grupo, el sistema envía un mensaje informativo a todos los usuarios notificando este hecho.

Sintaxis: NewGroup <Grupo>

Añadirse a un grupo de usuarios: El usuario se une a un grupo de usuarios. Esto provoca que el usuario se subscriba al grupo y a partir de ese momento, reciba todos los mensajes enviados a ese grupo.

Sintaxis: JoinGroup <Grupo>

- Cerrar la aplicación: Desconecta el usuario del sistema (si está conectado) y finaliza la aplicación.

Sintaxis: Exit



Evaluación

La evaluación de la práctica se realizará en función de la funcionalidad implementada y de la calidad del sistema distribuido diseñado.

Un aspecto que se tendrá en cuenta es que la aplicación distribuida sea abierta, de forma que cada una de las aplicaciones desarrollada por los diferentes grupos de estudiantes sean compatibles entre sí. Para ello los diferentes grupos de prácticas/versiones de la aplicación tendrán que definir un interface común del objeto distribuido de forma que sean compatibles. Este interface común afectará a los métodos remotos necesarios para soportar la funcionalidad y los parámetros de estos, pero no a su implementación que puede/debe variar de un grupo a otro.

Se utilizarán los siguientes criterios a la hora de evaluar la práctica:

- Aspectos que se evalúan para cada una de las versiones:
 - Versión no distribuida (Susp)
 - Login/Logout/Exit (+1.0p)
 - Envío/Recepción mensajes a usuarios individuales (+1.0p)
 - Gestión Grupos (NewGroup/JoinGroup) (+1.0p)
 - Envío/Recepción mensajes a grupos (+1.5p)
 - Envío notificaciones del sistema (+1.0p)
 - Utilización callbacks (+1.0p)
 - Sistema abierto, compatible con el resto de versiones (+1.5p)
 - Utilización stubs dinámicos (+0.5p)
 - Seguridad aplicación (+0.5p)
 - Utilización JNDI (+0.5p)
- Evaluación del informe de la práctica:
 - + Informe completo, con diseño, validación (local/remota) realizada de la aplicación y descripción de los problemas encontrados. (+1.0p)
- Estilo del código.
 - Comentarios
 - Utilizar una correcta indentación
 - Descomposición Funcional (Código modular y estructurado)
 - Control de errores.
 - Test unitarios para verificar la correcta ejecución de la lógica de la aplicación.

Se puede tener en cuenta criterios adicionales en función de la implementación entregada.



Formato de entrega

MUY IMPORTANTE: La entrega de código que no compile correctamente, implicará suspender TODA la práctica.

No se aceptarán prácticas entregadas fuera de plazo (salvo por razones muy justificadas).

La entrega presencial de esta práctica es obligatoria para todos los miembros del grupo.

Comenzar vuestros programas con los comentarios:

```
/* -----  
Práctica 1.  
Código fuente: MsgRMI.java  
Grau Informàtica  
NIF i Nombre completo autor1.  
NIF i Nombre completo autor2.  
----- */
```

Para presentar la práctica dirigiros al apartado de Actividades del Campus Virtual de la asignatura de Sistemas Concurrentes y Paralelos, ir a la actividad "Práctica 1" y seguid las instrucciones allí indicadas.

Se creará un fichero tar/zip con todos los ficheros fuente de la práctica, con el siguiente comando:

```
$ tar -zcvf prac1.tgz fichero1 fichero2 ...
```

se creará el fichero "prac1.tgz" en donde se habrán empaquetado y comprimido los ficheros "fichero1", fichero2, y ...

Para extraer la información de un fichero tar se puede utilizar el comando:

```
$ tar -zxvf tot.tgz
```

El nombre del fichero tar tendrá el siguiente formato: "Apellido1Apellido2PRA2.tgz". Los apellidos se escribirán sin acentos. Si hay dos autores, indicar los dos apellidos de cada uno de los autores separados por "_". Por ejemplo, el estudiante "Perico Pirulo Palotes" utilizará el nombre de fichero: PiruloPalotesPRA1.tgz.



Fecha de entrega

Entrega a través de Sakai el 16 de octubre, entrega presencial en la clase de grupo de laboratorio durante la semana del 28 de octubre al 27 de Marzo de 2020.

