

1. Especificación de la práctica

En el desarrollo de esta práctica, los alumnos aprenderán a diseñar y a programar protocolos de comunicación. La práctica se basa en un sistema de chat al que denominaremos *nanoChat*, y que se describe a continuación. El sistema *nanoChat* está formado por un servidor de chat y un conjunto de clientes que se conectan al chat.

Un aspecto importante del sistema *nanoChat* es que la dirección de los servidores de chat se obtiene tras preguntar a un servidor de directorio. Los servidores de chat deben registrarse primero ante el directorio para registrar su dirección de red y posteriormente los clientes pueden buscar en el directorio cuáles son los servidores de chat disponibles. Una característica relevante del sistema es que el servidor de chat dispone de *salas* donde los clientes acceden para conversar con el resto de clientes que están en dicha *sala*. La Figura 1 muestra un esquema general orientativo del conjunto de componentes de *nanoChat*.

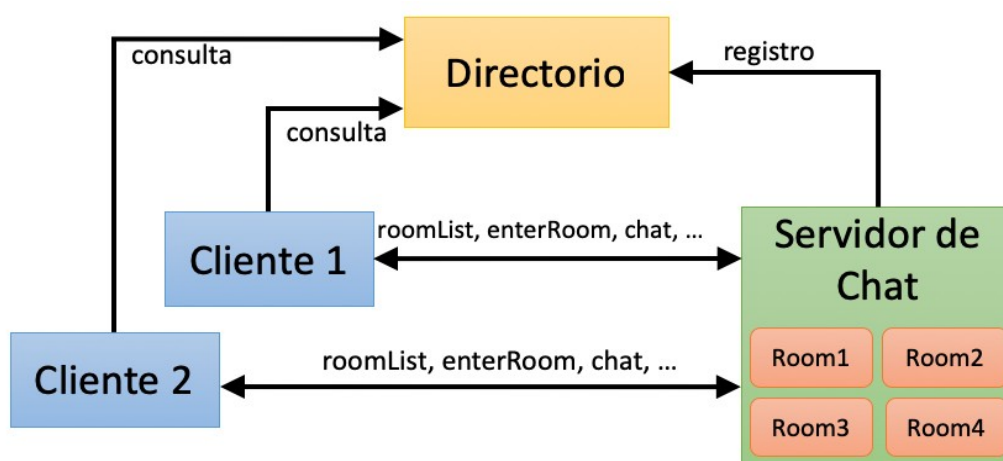


Figura 1. Elementos de *nanoChat*.

El objetivo global de esta práctica es que los alumnos sean capaces de diseñar los protocolos necesarios para intercambiar los mensajes involucrados así como de implementar el software de todos los elementos que constituyen el sistema *nanoChat*.

2. Diseño de los protocolos necesarios

Específicamente, los alumnos deberán llevar a cabo **el diseño de dos protocolos de comunicación de nivel de aplicación** del sistema *nanoChat*:

- Para la comunicación entre el cliente (o servidor de chat) y el directorio, los alumnos deberán diseñar un **protocolo confiable semi-dúplex basado en parada y espera**, que operará sobre un protocolo de nivel de transporte no confiable como UDP.
- Para la comunicación entre el cliente y el servidor de chat, los alumnos diseñarán un **protocolo asumiendo que el protocolo de nivel de transporte es confiable** (como TCP).

En particular, los alumnos deberán realizar:

- Respecto a los autómatas, se pide especificar, por separado, los siguientes autómatas:
 - Los autómatas relacionados con el servicio de directorio

- Autómata para el cliente cuando se comunica con el servidor de chat.
- Autómata para el servidor de chat cuando se comunica con el cliente.
- Respecto a las especificaciones de los mensajes a intercambiar, se deben utilizar los siguientes lenguajes:
 - Binario para el protocolo de comunicación con el directorio.
 - ASCII (Field:Value o Lenguaje de Marcas) para la comunicación entre cliente y servidor de chat.

Se deja a elección de los alumnos las decisiones relativas a los diferentes tipos de mensajes necesarios así como a la codificación de los distintos campos.

3. Implementación del software

En paralelo a la fase de diseño, los alumnos tendrá que implementar los elementos software necesarios para la práctica. Hay un material de partida que se proporciona a los alumnos:

- Esqueleto de código del servidor de directorio
- Implementación sin terminar del programa NanoChat (cliente), incluyendo:
 - Interfaz de usuario, basada en un sencillo intérprete de comandos (ya implementado).
 - Lógica del programa cliente (parcialmente implementado).
- Implementación sin terminar del programa NanoChatServer (servidor de chat), incluyendo:
 - Proceso principal que correrá en segundo plano y creará nuevos Threads a partir de solicitudes entrantes (ya implementado).
 - Hilo que atiende a cada cliente (sólo esbozado).
 - Gestor global del servidor (sólo esbozado).
 - Gestor de sala (sólo clase abstracta).
- Clases auxiliares:
 - Para la representación de mensajes (sólo esbozada).
 - Para la representación del estado de una sala (sólo esbozada).

Así pues, el trabajo de los alumnos consistirá en completar tanto el programa cliente como los servidores para conseguir que la práctica sea funcional.

3.1. Lógica del programa *Directory*

El programa *Directory* quedará a la espera, una vez lanzado, de los mensajes que vaya recibiendo, registrando los servidores de chat disponibles y respondiendo a consultas de los clientes. El servidor de directorio utilizará el puerto **6868/udp** para recibir mensajes entrantes. Este es el parámetro que podría especificarse al ejecutarlo:

- `-loss <probability>`: este parámetro indica cuál es la probabilidad de que se pierda un paquete en la comunicación con el directorio. Oscila entre 0 (no se pierde ninguno) y 1 (se pierden todos). Si no se indica la probabilidad de pérdida será 0. Los alumnos deberán programar un mecanismo por el cual, en el caso de especificar `-loss`, los paquetes recibidos por el directorio sean descartados con la probabilidad indicada.

Con la finalidad de que los alumnos puedan depurar el correcto funcionamiento de los clientes le solicitan servicio, el directorio debería imprimir por consola un breve resumen de los mensajes recibidos y enviados, así como de aquellos mensajes que se han perdido.

3.2 Lógica del programa *NanoChatServer* (servidor de chat)

El programa *NanoChatServer* quedará a la espera, una vez lanzado, de los mensajes que va recibiendo y va generando los hilos correspondientes para que atiendan a los clientes conectados. El servidor utilizará el puerto **6969/tcp** para atender peticiones de los clientes. Recibe este parámetro al ejecutarse:

- `<DirectoryServer>`: dirección en la que se encuentra el servidor de directorio.

Con la finalidad de que los alumnos puedan depurar el correcto funcionamiento de sus servidores, se sugiere que se imprima un resumen de los mensajes recibidos y enviados.

3.3 Lógica del programa *NanoChat* (cliente)

El programa *NanoChat* interactuará con el usuario mediante una línea de comandos. Al ejecutarlo se le debe suministrar un argumento:

- `<DirectoryServer>`: dirección en la que se encuentra el servidor de directorio.

IMPORTANTE: Una vez en ejecución, el programa cliente ya acepta algunas órdenes. No obstante, ninguna de dichas órdenes está programada, es decir, no hay ningún efecto al teclearlas (excepto `help`) puesto que dicha funcionalidad debe ser añadida por los estudiantes.

Estas son las órdenes que se reconocen:

- `nick <nickname>` (para establecer el nick del usuario en el servidor)
- `roomlist` (para obtener la lista de salas del servidor)
- `enter <room>` (para solicitar entrar en la sala indicada por parámetro)
- `send <text>` (para enviar un mensaje al chat de la sala)
- `exit` (para salir de la sala)
- `info` (para mostrar información de la sala)
- `quit` (cierra la conexión con el servidor y termina el programa)
- `help` (muestra información sobre las órdenes válidas)

La secuencia correcta en la cual se pueden teclear dichas órdenes, o las situaciones en las que algunas de ellas no están permitidas, dependerá del autómata que los alumnos hayan diseñado. Por tanto, será responsabilidad de los mismos controlar dicho orden.

NOTA: Los estudiantes, tomando como base el código de la clase `NCSHELL`, pueden modificar el shell a su gusto si lo consideran conveniente. Se trata simplemente de un material de partida.

3.4 Funcionalidad obligatoria a implementar por los alumnos

Ninguno de los programas está concluido, forma parte de la práctica terminar su funcionalidad. Para ello tendrán que llevar a cabo, entre otras, las siguientes tareas:

- Implementar el formato de mensajes diseñado para que los segmentos UDP intercambiados

con el servidor de directorio contengan mensajes sintácticamente correctos.

- Implementar el formato de mensajes diseñado para que los segmentos TCP intercambiados entre cliente y servidor contengan mensajes adecuadamente formateados.
- El servidor de chat debe registrar en el directorio su dirección de red.
- El cliente debe recuperar del servidor de directorio la dirección del servidor de chat al que conectar.
- Al registrar el nick del usuario en el servidor de chat hay que verificar si no está duplicado.
- Obtener y mostrar la lista de salas disponibles en un servidor de chat.
- Solicitar acceso a una sala y conocer el resultado de dicha solicitud.
- Obtener la información de una sala.
- Permitir el envío de mensajes de texto a todos los clientes conectados a una misma sala.
- Permitir que el usuario pueda salir de la sala actual y entrar en otra.
- Permitir que el usuario pueda desconectarse del servidor.
- El programa NanoChatServer deberá estar funcionando en todo momento en segundo plano.

La implementación de esta funcionalidad básica permite alcanzar una calificación de hasta un máximo de 8 puntos.

3.4 Funcionalidad opcional a implementar (mejoras)

Para obtener una mayor calificación en la práctica se propone la siguiente funcionalidad adicional que podría implementarse, aunque se valorarán también de forma positiva otras mejoras que los alumnos quieran plantear al profesorado. Se indica también la puntuación adicional que podría obtenerse con la implementación de cada una de ellas:

Mejora	Puntuación máxima
Notificar a los demás usuarios de la sala las entradas/salidas de otros usuarios en las salas conforme se vayan produciendo	0,5
La posibilidad de renombrar una sala.	0,5
La posibilidad de crear nuevas salas en el servidor	0,5
Enviar un mensaje privado a un usuario de la sala (sin que los demás puedan verlo).	1
Ver el histórico de mensajes enviados a la sala.	1
Definir administradores de sala. Que los administradores de sala puedan echar o ascender a otros usuarios.	1
Enviar ficheros binarios a través de una sala.	1

4. Detalles de la entrega

El trabajo que los alumnos deberán desarrollar es el siguiente:

- Programa cliente, servidor de directorio y servidor de chat.
- Documentación de la práctica.

Instrucciones detalladas:

- Las prácticas deben ser realizadas obligatoriamente por grupos de dos personas.
- Los grupos deberán subir al Aula Virtual, a la tarea denominada “Práctica de NanoChat” un archivo comprimido .ZIP que contenga un documento PDF con los siguientes apartados:
 - Introducción.
 - Diseño de los protocolos.
 - Formato de los mensajes (de directorio y de chat) y ejemplos de los mismos.
 - Autómatas de cliente de chat, servidor de chat y de servidor de directorio.
 - Detalles sobre los principales aspectos de implementación
 - Implementación del formato de los mensajes.
 - Mecanismo de gestión de salas.
 - Mejoras adicionales implementadas
 - Conclusiones.
- Dicho archivo comprimido incluirá el proyecto con la implementación de todo el código. Debe ser un proyecto Eclipse listo para ser importado y ejecutado.
- También se incluirán dos archivos .jar que permitan ejecutar tanto el programa cliente como servidor.
- El nivel del compilador utilizado debe ser al menos Java 1.7.
- Las prácticas deben funcionar en los laboratorios, en el sistema operativo Ubuntu.

El día tope de entrega será el día 19 de mayo.

Las entrevistas se desarrollaran durante las horas de prácticas habituales, la semana siguiente a la entrega. Los alumnos deberán asistir a las clases correspondientes al subgrupo donde se encuentren ubicados. En caso de encontrarse en distintos subgrupos, elegirán el día en el que ambos miembros puedan estar presentes.

5. Rúbrica del diseño de los protocolos

Quizás ya conozcáis lo que es una rúbrica, pero en todo caso os dejamos las respuestas a algunas preguntas que os pueden servir para conocer qué es una rúbrica y sus beneficios en los procesos de evaluación del aprendizaje.

5.1. ¿Qué es una rúbrica?

Una rúbrica es un recurso para la evaluación y calificación objetiva del aprendizaje, de los conocimientos o del desempeño de los estudiantes en una actividad concreta y que establece **criterios** y una **escala de valoración** para cada uno de ellos.

En general las rúbricas son presentadas en forma de tabla o matriz donde se asocia a cada criterio su escala de valoración.

5.2. ¿En qué me beneficia el uso de las rúbricas?

Las rúbricas establecen los criterios de evaluación de una manera objetiva, pública, transparente y consistente. Esto permite a los estudiantes conocer sobre qué aspectos deben centrar sus esfuerzos para ser valorados positivamente en la realización de la tarea que se trate.

A continuación detallamos las rúbricas para los entregables que se definen en las diferentes etapas del proyecto.

5.3. Rúbrica de la práctica de diseño

Una rúbrica está formada habitualmente por más de un criterio, y cada criterio dispone de un peso dentro de la calificación de la tarea. En el caso concreto de la práctica de diseño que se enuncia en este documento, la rúbrica que tiene asociada se puede ver en la siguiente tabla:

Criterio	Peso	Sobresaliente (2)	Alcanzado (1)	Insuficiente (0)
Idoneidad y claridad de la representación de los autómatas	40%	Los autómatas son correctos, completos y legibles. Soportan toda la funcionalidad solicitada y sus figuras son perfectamente legibles en el documento.	Los autómatas de son legibles, aunque contienen algún fallo menor (ausencia de una funcionalidad). Las figuras del documento son legibles.	Los autómatas no reflejan correctamente varias de las funcionalidades y además su legibilidad es escasa.

Criterio	Peso	Sobresaliente (2)	Alcanzado (1)	Insuficiente (0)
Adecuación de los formatos de representación de los mensajes	40%	Se representan correctamente todos los mensajes que aparecen en los autómatas. Se detalla la codificación de los campos, su longitud, orden (en su caso) y además se incluyen ejemplos de intercambios de mensajes usando el formato de los mismos.	Se representan correctamente la mayoría de los mensajes que aparecen en los autómatas. Se detalla la mayor parte de la codificación de los campos, longitud, orden (en su caso) y se incluyen ejemplos.	La mayor parte de los mensajes no han sido representados de forma correcta. Los detalles sobre sus campos son insuficientes y no hay ejemplos de intercambios.
Corrección de la documentación presentada	20%	El documento contiene una portada, índice de contenidos, número de páginas, respeta la estructura indicada, es claro y carece de faltas ortográficas y gramaticales.	El documento está estructurado aunque carece de ciertos aspectos formales. La estructura es la correcta y carece de faltas ortográficas y gramaticales.	El documento carece de estructura y no está redactado correctamente.

5.4. ¿Cómo se califica en función de una rúbrica?

Si suponemos que un estudiante ha obtenido un nivel 2 para el primer criterio, un nivel 1 para el segundo criterio y un nivel 1 para el tercer criterio, la nota sería:

$$\text{Calificación final} = (2 \cdot 40\% + 1 \cdot 40\% + 1 \cdot 20\%) / 2 = 140\% / 2 = 70\% = 7$$