

UNIDAD DE APRENDIZAJE : Aplicaciones para comunicaciones en red UNIDAD TEMÁTICA II: Sockets de datagrama	
No. Y Título de la práctica: <p style="text-align: center;">Práctica no. 4 Servicio de chat</p>	Tiempo de realización: 3 horas
Objetivo de la práctica: El estudiante implementará un servicio de chat haciendo uso de sockets de datagrama, así como sockets de multidifusión	
<p>Situación problemática: Muchas empresas hacen uso de Internet para ofrecer sus productos y servicios, ya que a través de este medio pueden estar en contacto directo con los clientes potenciales. Algunas de las principales ventajas de Internet como medio para realizar negocios son la cobertura a nivel global, así como la disponibilidad del servicio los 365 días, las 24 hrs. del día. Sin embargo, estas ventajas se convierten también en un reto para las empresas, pues se debe proporcionar a los clientes una vía de comunicación directa para resolver las dudas de los clientes potenciales, atender sus peticiones, así como proporcionarles información adicional a la expuesta en sus portales. Existen diversos recursos que pueden ser utilizados, como los foros o el correo electrónico, pero ninguno de ellos permite la comunicación en tiempo real. El chat es un excelente recurso para este tipo de propósito, ya que además de ser una vía de comunicación en tiempo real, permite comunicar a dos o más usuarios entre sí. Típicamente los chats se han implementado haciendo uso de sockets de flujo, pero existen otras alternativas que también merece la pena probar, tales como los sockets de datagrama o los de multidifusión</p> <p>¿Qué ventajas tendrá la implementación de un servicio de chat utilizando sockets de datagrama o de multidifusión?</p>	
<p>Competencia específica: Desarrolla aplicaciones en red, con base en el modelo cliente-servidor y utilizando la interfaz de sockets de datagrama, así como sockets de multidifusión bloqueantes, para el envío de datos.</p>	
<p>Competencias genéricas:</p> <ul style="list-style-type: none"> • Aplica los conocimientos en la práctica • Demuestra habilidad para trabajar en equipo • Demuestra capacidad de investigación • Desarrolla aplicaciones en red con base en la tecnología más adecuada 	<p>Elementos de competencia:</p> <ul style="list-style-type: none"> • Programa aplicaciones en red con base en el modelo Cliente-Servidor y la interfaz de aplicaciones de sockets de multidifusión • Analiza los servicios definidos en la capa de transporte • Emplea el modelo Cliente-Servidor para construir aplicaciones en red

Introducción

Hoy día, en muchos tipos de aplicaciones se cuenta con un servicio de chat, ya sea con fines de diversión o de negocio. Este tipo de servicio es muy socorrido cuando se trata de brindar una comunicación más personalizada que la brindada por medios tales como correo electrónico, o foros. En el chat la comunicación puede ser fluida, es decir, en tiempo real y además pueden interactuar dos o más personas a la vez.

Recursos y/o materiales

- Manual de prácticas de laboratorio de Aplicaciones para Comunicaciones en Red
- Plumones
- Bibliografía
- Internet
- Computadora
- IDE de desarrollo
- Apuntes

Instrucciones

En esta práctica debes implementar una aplicación chat que permita a los usuarios comunicarse entre sí a través de una sala común, así como mensajes privados haciendo uso de sockets de multidifusión. La aplicación también permitirá a los usuarios enviar emoticones a otros usuarios. El formato de los mensajes a utilizar por parte de la aplicación será el siguiente:

Mensaje	Significado
<inicio> Juan	El usuario Juan ha entrado al chat
<msj><Juan> Hola a todos	El usuario Juan escribe un mensaje en la sala comunal
<privado><Juan><Laura>Cómo estás?	El usuario Juan Le envía un mensaje privado al usuario Laura
<msj><Juan> Saludos :)	El usuario Juan envía un mensaje con un emoticon, el cual será interpretado por la aplicación cliente y sustituido por una imagen ☺

Desarrollo de la práctica

A partir de los programas ClienteMulticast y ServidorMulticast, o multicastC-JAVA que te serán proporcionados por el profesor deberás realizar las siguientes modificaciones:

- ❖ La lógica del programa estará implementada en el lado del cliente.
- ❖ El programa ServidorMulticast deberá unirse a la dirección de grupo 230.1.1.1 y usar el puerto 4000 para leer todos los mensajes enviados por los usuarios de la sala común.
- ❖ Cuando un usuario ejecute la aplicación cliente, éste deberá proporcionar un nombre de usuario, entonces se enviará un mensaje a la dirección de grupo 230.1.1.1 al puerto destino 4000 con el mensaje "<inicio>[nombre-usuario]". Todos los clientes del chat deberán actualizar su lista de usuarios en línea al recibir el mensaje.
- ❖ Una vez enviado el mensaje de inicio, la aplicación cliente deberá iniciar la lectura del socket para recibir posibles mensajes de otros clientes.
- ❖ Cuando llegue un mensaje, este deberá ser interpretado en base al tipo de mensaje y en su caso, deberán realizarse las acciones correspondientes.
- ❖ En caso de que el mensaje recibido sea un mensaje privado, solamente la aplicación cliente que tenga asociado el nombre de usuario del destinatario indicado en el mensaje será quien muestre dicho mensaje en pantalla. Los demás clientes ignorarán esos mensajes.
- ❖ Cuando el mensaje recibido contenga emoticones, estos deberán ser interpretados mostrándose las imágenes correspondientes en pantalla.

Cierre de la práctica

Preguntas:

1. ¿Qué ventajas presenta el uso de sockets de multidifusión contra unidifusión?
2. ¿Qué modificaciones a nivel del campo Tiempo de vida (TTL) es necesario considerar?
3. ¿Qué consecuencia tendrá en el desempeño de la aplicación habilitar el algoritmo de Nagle ?