|  |  |  |
| --- | --- | --- |
|  | **INSTITUTO POLITECNICO NACIONAL**  Escuela Superior de Cómputo |  |

Alumnos:

Guerra Vargas Irving Cristobal

Pantoja Flores Pablo

Unidad de Aprendizaje: Aplicaciones para Comunicaciones en Red

Profesor: Axel Ernesto Moreno Cervantes

**PRACTICA 3: Chat**

Grupo: 3CM7

Contenido

[Introducción 3](#_Toc516597762)

[Desarrollo 3](#_Toc516597763)

[Códigos 6](#_Toc516597764)

[Conclusiones 16](#_Toc516597765)

# Introducción

Los sockets multicast son de suma importancia, puesto que nos permiten enviar un mensaje por red y que este sea recibido por otros ordenadores simultáneamente.

Para estos sockets se utilizan, como su nombre lo dice, direcciones multicast, es decir direcciones en el rango 224.0.0.0 a 239.255.255.255.

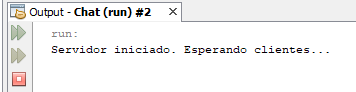
Una de las aplicaciones más usadas en el mundo son las aplicaciones de mensajería o chat.

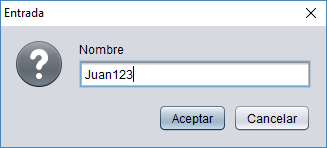
Vemos que hoy en día todas esas aplicaciones tienen funciones como el envío de archivos, emoticonos, encuestas, la creación de grupos de conversación, entre otras funcionalidades.

En esta práctica se utilizarán los sockets multicast con el fin de simular el funcionamiento de un chat.

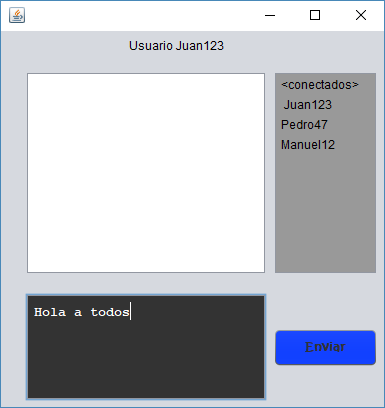
# Desarrollo

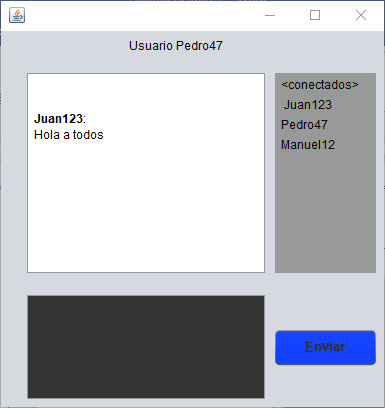
La aplicación comienza solicitando al usuario un nombre con el cual identificarse dentro del grupo de conversación.

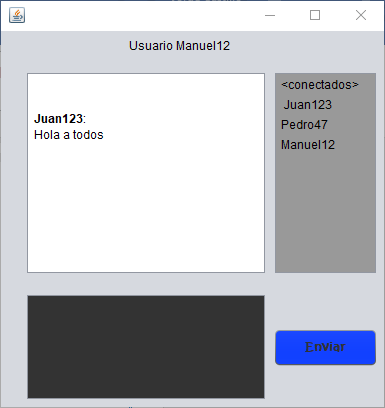


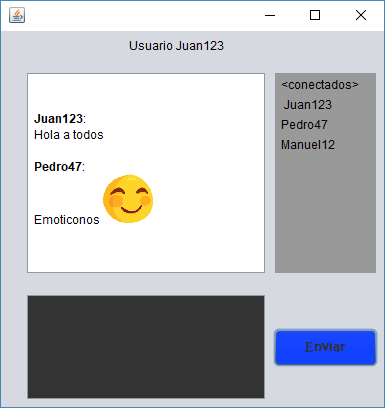


Después se le presenta a este un pequeño frame mediante el cual podrá enviar mensajes a otros usuarios que se encuentren conectados en ese momento. Es posible también enviar mensajes privados a un usuario en específico.









# Códigos

**Chat.java**

**package chat;**

**import java.awt.event.KeyEvent;**

**import java.awt.event.MouseAdapter;**

**import java.awt.event.MouseEvent;**

**import java.util.ArrayList;**

**import java.util.List;**

**import javax.swing.DefaultListModel;**

**import javax.swing.JList;**

**import javax.swing.JOptionPane;**

**public class Chat extends javax.swing.JFrame {**

**private String user;**

**Client cliente;**

**String[] conectados;**

**String textoChat = "";**

**List<String> listaConectados;**

**String privateTo = "";**

**DefaultListModel modelo;**

**public Chat() {**

**initComponents();**

**txtChat.setContentType("text/html");**

**txtChat.setEditable(false);**

**txtChat.setText("");**

**listaConectados=new ArrayList<>();**

**cliente = new Client(this);**

**do{**

**user = JOptionPane.showInputDialog(this, "Nombre");**

**}while(user.equals("") || user.equals(" ") || user == null || user.length() < 4);**

**cliente.start();**

**cliente.envia("<inicio>"+user);**

**cliente.envia("<msg> <b>"+user+"</b> se ha conectado.");**

**txtEstado.setText("¡Hola! "+user);**

**listConectados.addMouseListener(new MouseAdapter(){**

**public void mouseClicked(MouseEvent evt) {**

**JList list =(JList)evt.getSource();**

**if (evt.getClickCount() == 2) {**

**int index = list.locationToIndex(evt.getPoint());**

**privateTo = (String)modelo.getElementAt(index);**

**String texto = JOptionPane.showInputDialog("Mensaje privado a " + privateTo);**

**if(privateTo != user){**

**if(texto!=null){**

**cliente.envia("<privado><"+user+"><"+privateTo+"> "+texto);**

**}**

**}**

**} else if (evt.getClickCount() == 3) { // Triple-click**

**int index = list.locationToIndex(evt.getPoint());**

**privateTo = (String)modelo.getElementAt(index);**

**}**

**}**

**});**

**}**

**public void updateUsersList(String usersCSV){**

**String[] users = usersCSV.split(",");**

**modelo = new DefaultListModel();**

**listConectados.removeAll();**

**for(String user:users){**

**user.replace(" ","");**

**user.trim();**

**if(!user.contains("<conectados>") || user.isEmpty()){**

**modelo.addElement(user);**

**}**

**}**

**listConectados.setModel(modelo);**

**}**

**public void Agregar\_conversacion(String msg){**

**String senderUser="";**

**String toUser="";**

**System.out.println(msg);**

**msg = msg.replaceAll(":D", "<img width=\"50\" height=\"50\" src=\"file:./feliz.png\"></img>");**

**msg = msg.replaceAll("<3", "<img width=\"50\" height=\"50\" src=\"file:./enamorado.png\"></img>");**

**msg = msg.replaceAll(":c", "<img width=\"50\" height=\"50\" src=\"file:./llorar.png\"></img>");**

**msg = msg.replaceAll(":3", "<img width=\"50\" height=\"50\" src=\"file:./tierna.png\"></img>");**

**msg = msg.replaceAll("@gif1", "<img width=\"50\" height=\"50\" src=\"file:./michael.gif\"></img>");**

**msg = msg.replaceAll("@gif2", "<img width=\"50\" height=\"50\" src=\"file:./mario.gif\"></img>");**

**msg = msg.replaceAll("@gif3", "<img width=\"50\" height=\"50\" src=\"file:./pajaro.gif\"></img>");**

**if(msg.contains("<msg>")){**

**senderUser=msg.split("<")[2].split(">")[0];**

**if(senderUser.equals("b"))**

**textoChat=textoChat+"<br/><br/>"+msg;**

**else**

**textoChat=textoChat+"<br/><br/><b>"+senderUser+"</b>:<br/> "+msg;**

**txtChat.setText(textoChat);**

**}else if(msg.contains("<inicio>")){**

**msg.replace("<inicio>", "");**

**listaConectados.add(msg);**

**}**

**else if(msg.contains("<conectados")){**

**updateUsersList(msg);**

**}else if(msg.contains("privado")){**

**senderUser=msg.split("<")[2].replace(">", "");**

**toUser=msg.split("<")[3].split(">")[0];**

**if(toUser.equals(user))**

**{**

**textoChat=textoChat+"<br/><br/><b>Mensaje privado de "+senderUser+"</b>:<br/>"+msg;**

**txtChat.setText(textoChat);**

**}**

**}**

**int x;**

**txtChat.selectAll();**

**x = txtChat.getSelectionEnd();**

**txtChat.select(x,x);**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents**

**private void initComponents() {**

**jScrollPane2 = new javax.swing.JScrollPane();**

**listConectados = new javax.swing.JList();**

**jScrollPane3 = new javax.swing.JScrollPane();**

**txtEnvia = new javax.swing.JTextArea();**

**btnEnviar = new javax.swing.JButton();**

**txtEstado = new javax.swing.JLabel();**

**jScrollPane1 = new javax.swing.JScrollPane();**

**txtChat = new javax.swing.JEditorPane();**

**lblEmojiLlorar = new javax.swing.JLabel();**

**lblEmojiEnamorado = new javax.swing.JLabel();**

**lblEmojiFeliz = new javax.swing.JLabel();**

**lblEmojiTierno = new javax.swing.JLabel();**

**jLabel1 = new javax.swing.JLabel();**

**jLabel2 = new javax.swing.JLabel();**

**jLabel3 = new javax.swing.JLabel();**

**lblFondo = new javax.swing.JLabel();**

**setDefaultCloseOperation(javax.swing.WindowConstants.DO\_NOTHING\_ON\_CLOSE);**

**setBackground(new java.awt.Color(51, 51, 51));**

**addWindowListener(new java.awt.event.WindowAdapter() {**

**public void windowClosed(java.awt.event.WindowEvent evt) {**

**formWindowClosed(evt);**

**}**

**public void windowClosing(java.awt.event.WindowEvent evt) {**

**formWindowClosing(evt);**

**}**

**});**

**getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());**

**jScrollPane2.setViewportView(listConectados);**

**getContentPane().add(jScrollPane2, new org.netbeans.lib.awtextra.AbsoluteConstraints(430, 30, 150, 310));**

**txtEnvia.setColumns(20);**

**txtEnvia.setFont(new java.awt.Font("Century Gothic", 1, 18)); // NOI18N**

**txtEnvia.setLineWrap(true);**

**txtEnvia.setRows(5);**

**txtEnvia.addKeyListener(new java.awt.event.KeyAdapter() {**

**public void keyPressed(java.awt.event.KeyEvent evt) {**

**txtEnviaKeyPressed(evt);**

**}**

**});**

**jScrollPane3.setViewportView(txtEnvia);**

**getContentPane().add(jScrollPane3, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 540, 500, 70));**

**btnEnviar.setBackground(new java.awt.Color(255, 255, 255));**

**btnEnviar.setFont(new java.awt.Font("Century Gothic", 1, 36)); // NOI18N**

**btnEnviar.setForeground(new java.awt.Color(51, 51, 51));**

**btnEnviar.setText(">");**

**btnEnviar.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**btnEnviarActionPerformed(evt);**

**}**

**});**

**getContentPane().add(btnEnviar, new org.netbeans.lib.awtextra.AbsoluteConstraints(540, 480, 60, 130));**

**txtEstado.setText("Estado: Desconectado");**

**getContentPane().add(txtEstado, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 10, -1, -1));**

**jScrollPane1.setViewportView(txtChat);**

**getContentPane().add(jScrollPane1, new org.netbeans.lib.awtextra.AbsoluteConstraints(22, 30, 390, 310));**

**lblEmojiLlorar.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\llorar.png")); // NOI18N**

**lblEmojiLlorar.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**lblEmojiLlorarMouseClicked(evt);**

**}**

**});**

**getContentPane().add(lblEmojiLlorar, new org.netbeans.lib.awtextra.AbsoluteConstraints(20, 360, 120, 120));**

**lblEmojiEnamorado.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\enamorado.png")); // NOI18N**

**lblEmojiEnamorado.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**lblEmojiEnamoradoMouseClicked(evt);**

**}**

**});**

**getContentPane().add(lblEmojiEnamorado, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 360, 120, 120));**

**lblEmojiFeliz.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\feliz.png")); // NOI18N**

**lblEmojiFeliz.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**lblEmojiFelizMouseClicked(evt);**

**}**

**});**

**getContentPane().add(lblEmojiFeliz, new org.netbeans.lib.awtextra.AbsoluteConstraints(280, 360, -1, 120));**

**lblEmojiTierno.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\tierna.png")); // NOI18N**

**lblEmojiTierno.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**lblEmojiTiernoMouseClicked(evt);**

**}**

**});**

**getContentPane().add(lblEmojiTierno, new org.netbeans.lib.awtextra.AbsoluteConstraints(410, 370, 120, 110));**

**jLabel1.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\pajaro.gif")); // NOI18N**

**jLabel1.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jLabel1MouseClicked(evt);**

**}**

**});**

**getContentPane().add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(690, 240, -1, -1));**

**jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/chat/img/michael.gif"))); // NOI18N**

**jLabel2.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jLabel2MouseClicked(evt);**

**}**

**});**

**getContentPane().add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(620, 430, 280, 140));**

**jLabel3.setIcon(new javax.swing.ImageIcon("F:\\ESCOM\\7\\Redes2\\Corte 1\\Practica 3 - Chat\\Chat\\mario.gif")); // NOI18N**

**jLabel3.addMouseListener(new java.awt.event.MouseAdapter() {**

**public void mouseClicked(java.awt.event.MouseEvent evt) {**

**jLabel3MouseClicked(evt);**

**}**

**});**

**getContentPane().add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(640, 10, -1, -1));**

**lblFondo.setIcon(new javax.swing.ImageIcon(getClass().getResource("/chat/img/wall1.jpg"))); // NOI18N**

**lblFondo.setText("jLabel1");**

**getContentPane().add(lblFondo, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 0, 920, 630));**

**pack();**

**}// </editor-fold>//GEN-END:initComponents**

**private void btnEnviarActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_btnEnviarActionPerformed**

**// TODO add your handling code here:**

**String texto=txtEnvia.getText();**

**cliente.envia("<msg><"+user+">"+texto);**

**txtEnvia.setText("");**

**}//GEN-LAST:event\_btnEnviarActionPerformed**

**private void formWindowClosed(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowClosed**

**// TODO add your handling code here:**

**//cliente.envia("<fin>"+user);**

**}//GEN-LAST:event\_formWindowClosed**

**private void formWindowClosing(java.awt.event.WindowEvent evt) {//GEN-FIRST:event\_formWindowClosing**

**// TODO add your handling code here:**

**cliente.envia("<fin>"+user);**

**cliente.envia("<msg> <b>"+user+"</b> se ha desconectado.");**

**System.exit(0);**

**}//GEN-LAST:event\_formWindowClosing**

**private void lblEmojiLlorarMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_lblEmojiLlorarMouseClicked**

**txtEnvia.append(":c");**

**}//GEN-LAST:event\_lblEmojiLlorarMouseClicked**

**private void lblEmojiEnamoradoMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_lblEmojiEnamoradoMouseClicked**

**txtEnvia.append("<3");**

**}//GEN-LAST:event\_lblEmojiEnamoradoMouseClicked**

**private void lblEmojiFelizMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_lblEmojiFelizMouseClicked**

**txtEnvia.append(":D");**

**}//GEN-LAST:event\_lblEmojiFelizMouseClicked**

**private void lblEmojiTiernoMouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_lblEmojiTiernoMouseClicked**

**txtEnvia.append(":3");**

**}//GEN-LAST:event\_lblEmojiTiernoMouseClicked**

**private void txtEnviaKeyPressed(java.awt.event.KeyEvent evt) {//GEN-FIRST:event\_txtEnviaKeyPressed**

**if(evt.getKeyCode() == KeyEvent.VK\_ENTER){**

**String texto=txtEnvia.getText();**

**cliente.envia("<msg><"+user+">"+texto);**

**txtEnvia.setText("");**

**}**

**}//GEN-LAST:event\_txtEnviaKeyPressed**

**private void jLabel3MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel3MouseClicked**

**txtEnvia.append("@gif2");**

**}//GEN-LAST:event\_jLabel3MouseClicked**

**private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel1MouseClicked**

**txtEnvia.append("@gif3");**

**}//GEN-LAST:event\_jLabel1MouseClicked**

**private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {//GEN-FIRST:event\_jLabel2MouseClicked**

**txtEnvia.append("@gif1");**

**}//GEN-LAST:event\_jLabel2MouseClicked**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(Chat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(Chat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(Chat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(Chat.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Chat().setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify//GEN-BEGIN:variables**

**private javax.swing.JButton btnEnviar;**

**private javax.swing.JLabel jLabel1;**

**private javax.swing.JLabel jLabel2;**

**private javax.swing.JLabel jLabel3;**

**private javax.swing.JScrollPane jScrollPane1;**

**private javax.swing.JScrollPane jScrollPane2;**

**private javax.swing.JScrollPane jScrollPane3;**

**private javax.swing.JLabel lblEmojiEnamorado;**

**private javax.swing.JLabel lblEmojiFeliz;**

**private javax.swing.JLabel lblEmojiLlorar;**

**private javax.swing.JLabel lblEmojiTierno;**

**private javax.swing.JLabel lblFondo;**

**private javax.swing.JList listConectados;**

**private javax.swing.JEditorPane txtChat;**

**private javax.swing.JTextArea txtEnvia;**

**private javax.swing.JLabel txtEstado;**

**// End of variables declaration//GEN-END:variables**

**}**

**Client.java**

**package chat;**

**import java.io.\*;**

**import java.net.\*;**

**import java.util.Vector;**

**public class Client extends Thread{**

**public static final String MCAST\_ADDR = "227.0.0.1";**

**public static final int MCAST\_PORT = 9013;**

**public static final int DGRAM\_BUF\_LEN=512;**

**InetAddress group =null;**

**Chat elChat;**

**public Client(Chat chat){**

**elChat = chat;**

**System.out.println("Cliente iniciado.");**

**try{**

**group = InetAddress.getByName(MCAST\_ADDR);**

**}catch(UnknownHostException e){**

**e.printStackTrace();**

**System.exit(1);**

**}**

**}**

**public Boolean envia(String msg){**

**try{**

**MulticastSocket socketEnvio = new MulticastSocket(MCAST\_PORT);**

**socketEnvio.joinGroup(group); // se configura para escuchar el paquete**

**DatagramPacket packet = new DatagramPacket(msg.getBytes(),msg.length(),group,MCAST\_PORT);**

**//System.out.println("Enviando: " + msg+" con un TTL = "+socketEnvio.getTimeToLive());**

**socketEnvio.send(packet);**

**socketEnvio.close();**

**return true;**

**}catch(IOException e){**

**e.printStackTrace();**

**return false;**

**}**

**}**

**public void run(){**

**try{**

**group = InetAddress.getByName(MCAST\_ADDR);//intenta resolver la direccion**

**}catch(UnknownHostException e){**

**e.printStackTrace();**

**System.exit(1);**

**}**

**boolean salta = true;**

**try{**

**MulticastSocket socket = new MulticastSocket(MCAST\_PORT); //socket tipo multicast**

**socket.joinGroup(group);//se une al grupo**

**String mensaje="";**

**while(salta){**

**byte[] buf = new byte[DGRAM\_BUF\_LEN];//crea arreglo de bytes**

**DatagramPacket recv = new DatagramPacket(buf,buf.length);//crea el datagram packet a recibir**

**socket.receive(recv);// ya se tiene el datagram packet**

**//System.out.println("Host remoto: "+recv.getAddress());**

**//System.out.println("Puerto: "+ recv.getPort());**

**byte [] data = recv.getData();**

**mensaje=new String(data);**

**//System.out.println("Datos recibidos: " + mensaje); // se guarda en un vector los datos recibidos**

**elChat.Agregar\_conversacion(mensaje);**

**}**

**}catch(IOException e){**

**e.printStackTrace();**

**System.exit(2);**

**}**

**}**

**}**

**gifs.java**

**/\***

**\* To change this license header, choose License Headers in Project Properties.**

**\* To change this template file, choose Tools | Templates**

**\* and open the template in the editor.**

**\*/**

**package chat;**

**/\*\***

**\***

**\* @author alumno**

**\*/**

**public class gifs extends javax.swing.JFrame {**

**/\*\***

**\* Creates new form gifs**

**\*/**

**public gifs() {**

**initComponents();**

**}**

**/\*\***

**\* This method is called from within the constructor to initialize the form.**

**\* WARNING: Do NOT modify this code. The content of this method is always**

**\* regenerated by the Form Editor.**

**\*/**

**@SuppressWarnings("unchecked")**

**// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents**

**private void initComponents() {**

**jButton1 = new javax.swing.JButton();**

**setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);**

**getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());**

**jButton1.setText("Salir");**

**jButton1.addActionListener(new java.awt.event.ActionListener() {**

**public void actionPerformed(java.awt.event.ActionEvent evt) {**

**jButton1ActionPerformed(evt);**

**}**

**});**

**getContentPane().add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(350, 10, 80, -1));**

**pack();**

**}// </editor-fold>//GEN-END:initComponents**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed**

**this.dispose();**

**new Chat().setVisible(true);**

**}//GEN-LAST:event\_jButton1ActionPerformed**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

**java.util.logging.Logger.getLogger(gifs.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (InstantiationException ex) {**

**java.util.logging.Logger.getLogger(gifs.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (IllegalAccessException ex) {**

**java.util.logging.Logger.getLogger(gifs.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

**java.util.logging.Logger.getLogger(gifs.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);**

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new gifs().setVisible(true);**

**}**

**});**

**}**

**// Variables declaration - do not modify//GEN-BEGIN:variables**

**private javax.swing.JButton jButton1;**

**// End of variables declaration//GEN-END:variables**

**}**

**Server.java**

**package chat;**

**import java.net.\*;**

**import java.io.\*;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class Server extends Thread{**

**//Direccion clase D valida. Sera el grupo al que todos se uniran.**

**public static final String MCAST\_ADDR = "227.0.0.1";**

**public static final int MCAST\_PORT = 9013;**

**public static final int DGRAM\_BUF\_LEN = 512;**

**public List<String> conectados = new ArrayList<>();**

**public String conect = "";**

**private void addUser(String msg){**

**String[] x=msg.split(">");**

**conectados.add(x[1].trim());**

**conect="<conectados>,";**

**for(int y=0;y<conectados.size();y++){**

**if(y+1 == conectados.size()){**

**conect+=conectados.get(y);**

**}else{**

**conect+=conectados.get(y)+",";**

**}**

**}**

**conect.replace(" ", "");**

**}**

**public void deleteUser(String msg){**

**conectados.remove(msg.split(">")[1].trim());**

**conect="<conectados>,";**

**for(int y=0;y<conectados.size();y++){**

**if(y+1 == conectados.size()){**

**conect+=conectados.get(y);**

**}else{**

**conect+=conectados.get(y)+",";**

**}**

**}**

**conect.replace(" ", "");**

**}**

**public void run(){**

**String msg = ""; // se cambiara para poner la ip de la maquina con lo siguiente**

**InetAddress group = null;**

**try{**

**//msg=InetAddress.getLocalHost().getHostAddress();**

**group = InetAddress.getByName(MCAST\_ADDR); //se trata de resolver dir multicast**

**}catch(UnknownHostException e){**

**e.printStackTrace();**

**System.exit(1);**

**}**

**for(;;){**

**try{**

**byte[] buf = new byte[DGRAM\_BUF\_LEN];//crea arreglo de bytes**

**MulticastSocket socket = new MulticastSocket(MCAST\_PORT); //Creamos Socket multicast**

**socket.joinGroup(group); //Nos unimos a la direccion multicast. Se configura para escuchar el paquete**

**DatagramPacket recv = new DatagramPacket(buf,buf.length);//crea el datagram packet a recibir**

**socket.receive(recv);// ya se tiene el datagram packet**

**String inicio = new String(recv.getData());**

**if(inicio.contains("<inicio>")){**

**addUser(inicio);**

**}else if(inicio.contains("<fin>"))**

**deleteUser(inicio);**

**DatagramPacket packet = new DatagramPacket(conect.getBytes(),conect.length(),group,MCAST\_PORT);**

**//System.out.println("Enviando: " + conect+" con un TTL = "+socket.getTimeToLive());**

**socket.send(packet);**

**socket.close();**

**}catch(IOException e){**

**e.printStackTrace();**

**System.exit(2);**

**}**

**try{**

**Thread.sleep(1000\*5);**

**}catch(InterruptedException ie){}**

**}**

**}**

**public static void main(String[] args) {**

**try{**

**Server mc2 = new Server();**

**mc2.start();**

**System.out.println("[OK] Servidor iniciado. Esperando clientes...");**

**}catch(Exception e){**

**e.printStackTrace();**

**}**

**}**

**}**

# Conclusiones

En esta práctica se hizo uso de Sockets Multicast, los cuales son una herramienta que nos permite mandar a distintos clientes la misma información. La práctica fue difícil puesto que el enviar las imágenes o los emoticonos tenía que ser de manera personalizada. De igual forma los emoticonos se tuvieron que mandar como imágenes para no presentar otras complicaciones extra.

La práctica nos ayudó a comprender un poco más cómo funcionan las aplicaciones de mensajería en grupo y sin duda fue muy útil puesto que dichas herramientas son muy utilizadas en la actualidad.