public class Field {

private static void Restricion(java.awt.event.KeyEvent evt, int m, int M, int t) {

JTextField x = (JTextField) evt.getSource();

int k = (int) evt.getKeyChar();

if (k < m || k > M) {

evt.consume();

}

if (x.getText().length() == t) {

evt.consume();

}

if (x.getText().length() >= t-1 && k != 8 && k != 127) {

x.transferFocus();

}

}

private static void StrgToInt(String x, int i, int[] ip) {

if (!x.equals("")) {

ip[i] = Integer.valueOf(x);

} else {

ip[i] = -1;

}

}

private static void BinStrgToInt(String x, int i, int[] ip) {

if (!x.equals("")) {

ip[i] = Integer.parseInt(x, 2);

} else {

ip[i] = -1;

}

}

private static String IntToStrg(int i, int[] ip) {

if (ip[i] != -1) {

String x = String.valueOf(ip[i]);

for (int k = 0; x.length() < 3; k++) {

x = "0" + x;

}

return x;

}

return "";

}

protected static String IntToBinaryStrg(int i) {

if (i != -1) {

String x = Integer.toBinaryString(i);

for (int k = 0; x.length() < 8; k++) {

x = "0" + x;

}

return x;

}

return "";

}

protected static void GuardarDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4, int ip[]) {

StrgToInt(v1.getText(), 0,ip);

StrgToInt(v2.getText(), 1,ip);

StrgToInt(v3.getText(), 2,ip);

StrgToInt(v4.getText(), 3,ip);

}

protected static void GuardarBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4, int[] ip) {

BinStrgToInt(v1.getText(), 0,ip);

BinStrgToInt(v2.getText(), 1,ip);

BinStrgToInt(v3.getText(), 2,ip);

BinStrgToInt(v4.getText(), 3,ip);

}

protected static void abrirDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4, int[] ip) {

v1.setText(IntToStrg(0,ip));

v2.setText(IntToStrg(1,ip));

v3.setText(IntToStrg(2,ip));

v4.setText(IntToStrg(3,ip));

}

protected static void abrirBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4, int[] ip) {

v1.setText(IntToBinaryStrg(ip[0]));

v2.setText(IntToBinaryStrg(ip[1]));

v3.setText(IntToBinaryStrg(ip[2]));

v4.setText(IntToBinaryStrg(ip[3]));

}

protected static void Reset(int[] ip) {

for (int i = 0; i < ip.length; i++) {

ip[i] = -1;

}

}

public static void check(java.awt.event.FocusEvent evt) {

JTextField x = (JTextField) evt.getSource();

if(!x.getText().equals("")){

if (Integer.valueOf(x.getText()) > 255) {

JOptionPane.showMessageDialog(null, x.getText() + " no es una entrada válida. "

+ "Especifique un valor entre 0 y 255.",

"Entrada invalida", 0);

x.setText("255");

}

}

}

public static void RestricionDec(java.awt.event.KeyEvent evt) {

Restricion(evt, 48, 57, 3);

}

public static void RestricionBin(java.awt.event.KeyEvent evt) {

Restricion(evt, 48, 49, 8);

}

}

//----------------------------------------------------------------------------------------------------

public class ipField extends Field{

static int[] ip = new int[4];

public static void abrirBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

abrirBin(v1, v2, v3, v4,ip);

}

public static void abrirDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

abrirDec(v1, v2, v3, v4,ip);

}

public static void GuardarBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

GuardarBin(v1, v2, v3, v4,ip);

}

public static void GuardarDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

GuardarDec(v1, v2, v3, v4,ip);

}

public static void Reset(){

Reset(ip);

}

public static int getTipo(){

if(ip[0]<=127){

//clase A

return 0;

}else if(ip[0]<=191){

///clase B

return 1;

}else if(ip[0]<=223){

//clase C

return 2;

}else if(ip[0]<=239){

//clase D

return 3;

}else if(ip[0]<=255){

//clase E

return 4;

}

return -1;

}

public static void setMask(JTextField x,JTextField m1,JTextField m2,JTextField m3,JTextField m4,JTextField mb1,JTextField mb2,JTextField mb3,JTextField mb4,JTabbedPane panel, JButton gm){

if(!x.getText().equals("")){

if(ip[0]<=127){

//clase A

panel.setEnabled(true);

m2.setEditable(true);

m3.setEditable(true);

m4.setEditable(true);

mb2.setEditable(true);

mb3.setEditable(true);

mb4.setEditable(true);

gm.setEnabled(true);

m1.setText("255");

mb1.setText("11111111");

m2.setText("");

mb2.setText("");

m3.setText("");

mb3.setText("");

m4.setText("");

mb4.setText("");

}else if(ip[0]<=191){

///clase B

panel.setEnabled(true);

m2.setEditable(false);

m3.setEditable(true);

m4.setEditable(true);

mb2.setEditable(false);

mb3.setEditable(true);

mb4.setEditable(true);

gm.setEnabled(true);

m1.setText("255");

mb1.setText("11111111");

m2.setText("255");

mb2.setText("11111111");

m3.setText("");

mb3.setText("");

m4.setText("");

mb4.setText("");

}else if(ip[0]<=223){

//clase C

panel.setEnabled(true);

m2.setEditable(false);

m3.setEditable(false);

m4.setEditable(true);

mb2.setEditable(false);

mb3.setEditable(false);

mb4.setEditable(true);

gm.setEnabled(true);

m1.setText("255");

mb1.setText("11111111");

m2.setText("255");

mb2.setText("11111111");

m3.setText("255");

mb3.setText("11111111");

m4.setText("");

mb4.setText("");

}else if(ip[0]<=255){

//clase D o clase E

panel.setEnabled(false);

m2.setEditable(false);

m3.setEditable(false);

m4.setEditable(false);

mb2.setEditable(false);

mb3.setEditable(false);

mb4.setEditable(false);

gm.setEnabled(false);

m1.setText("");

mb1.setText("");

m2.setText("");

mb2.setText("");

m3.setText("");

mb3.setText("");

m4.setText("");

mb4.setText("");

}

}else{

panel.setEnabled(false);

m2.setEditable(false);

m3.setEditable(false);

m4.setEditable(false);

mb2.setEditable(false);

mb3.setEditable(false);

mb4.setEditable(false);

gm.setEnabled(false);

}

}

}

//--------------------------------------------------------------------------------------------------------------------------------

public class maskField extends Field{

static int[] mask=new int[4];

public static void abrirBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

abrirBin(v1, v2, v3, v4,mask);

}

public static void abrirDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

abrirDec(v1, v2, v3, v4,mask);

}

public static void GuardarBin(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

GuardarBin(v1, v2, v3, v4,mask);

}

public static void GuardarDec(JTextField v1, JTextField v2, JTextField v3, JTextField v4){

GuardarDec(v1, v2, v3, v4,mask);

}

public static void Reset(){

Reset(mask);

}

public static boolean isOK(){

Field x = new Field();

String mascara=IntToBinaryStrg(mask[0])+

IntToBinaryStrg(mask[1])+

IntToBinaryStrg(mask[2])+

IntToBinaryStrg(mask[3]);

for(int i=0; i<mascara.length();i++){

if(mascara.charAt(i)=='0'){

for(int j=i;j<mascara.length();j++){

if(mascara.charAt(j)!='0'){

return false;

}

}

}

}

return true;

}

}

//--------------------------------------------------------------------------------------------------------------------------------

public class FrmGenerarIP extends javax.swing.JFrame {

/\*\*

\* Creates new form frmGenerarIP

\*/

Random rand = new Random();

JTextField ip1, ip2, ip3, ip4;

JTextField bip1, bip2, bip3, bip4;

FrmPrin x;

public FrmGenerarIP(FrmPrin x, JTextField ip1, JTextField ip2,JTextField ip3,JTextField ip4,JTextField bip1, JTextField bip2,JTextField bip3,JTextField bip4) {

initComponents();

this.ip1=ip1;

this.ip2=ip2;

this.ip3=ip3;

this.ip4=ip4;

this.bip1=bip1;

this.bip2=bip2;

this.bip3=bip3;

this.bip4=bip4;

this.x = x;

}

private FrmGenerarIP() {

throw new UnsupportedOperationException("Not yet implemented");

}

/\*\*

\* 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">

private void initComponents() {

jComboBox1 = new javax.swing.JComboBox();

jComboBox2 = new javax.swing.JComboBox();

jLabel2 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

clase = new javax.swing.JComboBox();

tipo = new javax.swing.JComboBox();

jLabel1 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton3 = new javax.swing.JButton();

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));

jLabel2.setText("jLabel2");

jButton1.setText("jButton1");

jButton2.setText("jButton2");

setTitle("Generar IP");

setResizable(false);

clase.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Cualquiera", "A", "B", "C", "D", "E" }));

clase.addItemListener(new java.awt.event.ItemListener() {

public void itemStateChanged(java.awt.event.ItemEvent evt) {

claseItemStateChanged(evt);

}

});

tipo.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Cualquiera", "Publica", "Privada" }));

jLabel1.setText("Clase:");

jLabel3.setText("IP:");

jButton3.setText("OK");

jButton3.addActionListener(new java.awt.event.ActionListener() {

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

jButton3ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGap(0, 0, Short.MAX\_VALUE)

.addComponent(jButton3))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1)

.addComponent(jLabel3))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(tipo, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(clase, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(0, 14, Short.MAX\_VALUE)))

.addContainerGap())

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(20, 20, 20)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(clase, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(tipo, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addGap(18, 18, Short.MAX\_VALUE)

.addComponent(jButton3)

.addContainerGap())

);

pack();

}// </editor-fold>

private void setClase(){

switch (clase.getSelectedIndex()){

case 0:

ipField.ip[0]=rand.nextInt(254)+1;

break;

case 1:

ipField.ip[0]=rand.nextInt(127)+1;

break;

case 2:

ipField.ip[0]=rand.nextInt(64)+128;

break;

case 3:

ipField.ip[0]=rand.nextInt(32)+192;

break;

case 4:

ipField.ip[0]=rand.nextInt(16)+224;

break;

case 5:

ipField.ip[0]=rand.nextInt(15)+240;

break;

}

}

private void setTipo(){

switch (tipo.getSelectedIndex()){

case 1:

while((ipField.ip[0]==10) || ((ipField.ip[0]==172) && (ipField.ip[1]>=16 && ipField.ip[1]<=31)) || (ipField.ip[0]==192 && ipField.ip[1]==168)){

setClase();

}

break;

case 2:

int x;

if(clase.getSelectedIndex()==0){

x=rand.nextInt(3)+1;

}else{

x=clase.getSelectedIndex();

}

switch (x){

case 1:

ipField.ip[0]=10;

break;

case 2:

ipField.ip[0]=172;

ipField.ip[1]=rand.nextInt(16)+16;

break;

case 3:

ipField.ip[0]=192;

ipField.ip[1]=168;

break;

}

break;

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

setClase();

ipField.ip[1]=rand.nextInt(256);

ipField.ip[2]=rand.nextInt(256);

ipField.ip[3]=rand.nextInt(256);

if(tipo.isEnabled()){

setTipo();

}

ipField.abrirDec(ip1, ip2, ip3, ip4);

ipField.abrirBin(bip1, bip2, bip3, bip4);

this.setVisible(false);

x.setMask();

}

private void claseItemStateChanged(java.awt.event.ItemEvent evt) {

// TODO add your handling code here:

switch (clase.getSelectedIndex()){

case 0:

case 1:

case 2:

case 3:

tipo.setEnabled(true);

break;

case 4:

case 5:

tipo.setEnabled(false);

break;

}

}

//--------------------------------------------------------------------------------------------------------------------------------

public class FrmGenerarMask extends javax.swing.JFrame {

/\*\*

\* Creates new form FrmGenerarMask

\*/

JTextField ip1, ip2, ip3, ip4;

JTextField bip1, bip2, bip3, bip4;

public FrmGenerarMask(JTextField ip1, JTextField ip2,JTextField ip3,JTextField ip4,JTextField bip1, JTextField bip2,JTextField bip3,JTextField bip4) {

initComponents();

this.ip1=ip1;

this.ip2=ip2;

this.ip3=ip3;

this.ip4=ip4;

this.bip1=bip1;

this.bip2=bip2;

this.bip3=bip3;

this.bip4=bip4;

}

/\*\*

\* 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">

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

cant = new javax.swing.JTextField();

OK = new javax.swing.JButton();

setTitle("Generar Mascara");

setResizable(false);

addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

formComponentShown(evt);

}

});

jLabel1.setText("Cantidad de Host:");

cant.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

cantKeyTyped(evt);

}

});

OK.setText("OK");

OK.addActionListener(new java.awt.event.ActionListener() {

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

OKActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(OK)

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(cant, javax.swing.GroupLayout.PREFERRED\_SIZE, 71, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(cant, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(OK)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void OKActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(!cant.getText().equals("")){

int x=Integer.valueOf(cant.getText());

int max=0;

char clase='A';

switch (ipField.getTipo()){

case 0:

max=16777214;

clase ='A';

break;

case 1:

max=65534;

clase='B';

break;

case 2:

max=254;

clase='C';

break;

}

if(x>max){

JOptionPane.showMessageDialog(this, "Clase: "+clase+" no puede tener más de "+max+" Host", "Fuera de Rango", 0);

}else{

String mask="";

int i;

for (i=0; (Math.pow(2, i)-2)<x;i++){}

for (int j=0; j<i;j++){

mask+="0";

}

for (int j=0;mask.length()<24;j++){

mask="1"+mask;

}

String []vec=new String[3];

vec[0]="";

vec[1]="";

vec[2]="";

for (int j=0;j<8;j++){

vec[0]+=mask.charAt(j);

vec[1]+=mask.charAt(j+8);

vec[2]+=mask.charAt(j+16);

}

maskField.mask[0]=255;

maskField.mask[1]=Integer.parseInt(vec[0], 2);

maskField.mask[2]=Integer.parseInt(vec[1], 2);

maskField.mask[3]=Integer.parseInt(vec[2], 2);

maskField.abrirBin(bip1, bip2, bip3, bip4);

maskField.abrirDec(ip1, ip2, ip3, ip4);

this.setVisible(false);

}

}

}

private void cantKeyTyped(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

JTextField x = (JTextField) evt.getSource();

int k = (int) evt.getKeyChar();

if(k==10){

OKActionPerformed(null);

}

if (k < '0' || k > '9') {

evt.consume();

}

if (x.getText().length() == 8) {

evt.consume();

}

}

private void formComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

cant.setText("");

}

//------------------------------------------------------------------------------------------------------------------------

public class FrmInfo extends javax.swing.JFrame {

/\*\*

\* Creates new form FrmInfo

\*/

String ip;

String mask;

String ipRed;

String ipBroad;

int bits;

int clase;

int subRed;

frmSubNet sn;

frmHost h;

int [] vec;

int [] vec1;

int tipo;

DefaultListModel list;

public FrmInfo() {

initComponents();

vec=ipField.ip.clone();

vec1=maskField.mask.clone();

tipo=ipField.getTipo();

javax.swing.ToolTipManager.sharedInstance().setDismissDelay( 999999999 );

javax.swing.ToolTipManager.sharedInstance().setInitialDelay( 0 );

ip=ipField.ip[0]+"."+ipField.ip[1]+"."+ipField.ip[2]+"."+ipField.ip[3];

lIP.setText("IP: "+ip);

lIP.setToolTipText(Field.IntToBinaryStrg(ipField.ip[0])+" . "+

Field.IntToBinaryStrg(ipField.ip[1])+" . "+

Field.IntToBinaryStrg(ipField.ip[2])+" . "+

Field.IntToBinaryStrg(ipField.ip[3]));

this.setTitle(""+ip);

if(ipField.ip[0]<=127){

lClase.setText("Clase: A");

}else if(ipField.ip[0]<=191){

lClase.setText("Clase: B");

}else if(ipField.ip[0]<=223){

lClase.setText("Clase: C");

}else if(ipField.ip[0]<=239){

lClase.setText("Clase: D");

this.setSize(new Dimension(230, 90));

}else if(ipField.ip[0]<=255){

lClase.setText("Clase: E");

this.setSize(new Dimension(230, 90));

}

if( (ipField.ip[0]==10) || ((ipField.ip[0]==172) && (ipField.ip[1]>=16 && ipField.ip[1]<=31)) || (ipField.ip[0]==192 && ipField.ip[1]==168)){

lPublic.setText("IP Privada");

}

else{

lPublic.setText("IP Publica");

}

lMask.setText("Máscara: "+ maskField.mask[0]+"."+maskField.mask[1]+"."+maskField.mask[2]+"."+maskField.mask[3]);

lMask.setToolTipText(Field.IntToBinaryStrg(maskField.mask[0])+" . "+

Field.IntToBinaryStrg(maskField.mask[1])+" . "+

Field.IntToBinaryStrg(maskField.mask[2])+" . "+

Field.IntToBinaryStrg(maskField.mask[3]));

bits=0;

mask=Field.IntToBinaryStrg(maskField.mask[0])+

Field.IntToBinaryStrg(maskField.mask[1])+

Field.IntToBinaryStrg(maskField.mask[2])+

Field.IntToBinaryStrg(maskField.mask[3]);

for(int i=0;i<mask.length();i++){

if(mask.charAt(i)=='1'){

bits++;

}

}

lMaskBits.setText("Máscara de "+bits+" bits");

ipRed=(ipField.ip[0] & maskField.mask[0])+"."+

(ipField.ip[1] & maskField.mask[1])+"."+

(ipField.ip[2] & maskField.mask[2])+"."+

(ipField.ip[3] & maskField.mask[3]);

lIPRed.setText("Ip de Red: "+ ipRed);

lIPRed.setToolTipText(Field.IntToBinaryStrg(ipField.ip[0] & maskField.mask[0])+" . "+

Field.IntToBinaryStrg(ipField.ip[1] & maskField.mask[1])+" . "+

Field.IntToBinaryStrg(ipField.ip[2] & maskField.mask[2])+" . "+

Field.IntToBinaryStrg(ipField.ip[3] & maskField.mask[3]));

ipBroad= (ipField.ip[0] | ~(-256 | maskField.mask[0]))+"."+

(ipField.ip[1] | ~(-256 | maskField.mask[1]))+"."+

(ipField.ip[2] | ~(-256 | maskField.mask[2]))+"."+

(ipField.ip[3] | ~(-256 | maskField.mask[3]));

lIPBroad.setText("Ip de Broadcast: "+ ipBroad);

lIPBroad.setToolTipText(Field.IntToBinaryStrg(ipField.ip[0] | ~(-256 | maskField.mask[0]))+" . "+

Field.IntToBinaryStrg(ipField.ip[1] | ~(-256 | maskField.mask[1]))+" . "+

Field.IntToBinaryStrg(ipField.ip[2] | ~(-256 | maskField.mask[2]))+" . "+

Field.IntToBinaryStrg(ipField.ip[3] | ~(-256 | maskField.mask[3])));

if(ipField.getTipo()==0){

clase=128;

subRed=bits-8;

}else if (ipField.getTipo()==1){

clase=16384;

subRed=bits-16;

}else if (ipField.getTipo()==2){

clase=2097152;

subRed=bits-24;

}

lCantRed.setText("Cantidad de Redes: "+clase);

lCantSubRed.setText("<html><font color=\"#0000CF\"><u>"+"Cantidad de Sub Redes: "+(int)Math.pow(2, subRed)+"</u></font></html>");

lCantHost.setText("<html><font color=\"#0000CF\"><u>"+"Cantidad de Host por Red: "+(int)(Math.pow(2, 32-bits)-2)+"</u></font></html>");

if(ip.equals(ipRed)){

lIpTipo.setText("Ip de Red");

}else if(ip.equals(ipBroad)){

lIpTipo.setText("Ip de Broadcats");

}else{

lIpTipo.setText("Ip de Host");

}

}

/\*\*

\* 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">

private void initComponents() {

lMask = new javax.swing.JLabel();

lClase = new javax.swing.JLabel();

lPublic = new javax.swing.JLabel();

lIP = new javax.swing.JLabel();

lMaskBits = new javax.swing.JLabel();

lIPRed = new javax.swing.JLabel();

lIPBroad = new javax.swing.JLabel();

lCantRed = new javax.swing.JLabel();

lIpTipo = new javax.swing.JLabel();

lCantSubRed = new javax.swing.JLabel();

lCantHost = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setResizable(false);

lMask.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

lMask.setText("Máscara: 000.000.000.000");

lMask.setToolTipText("");

lClase.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lClase.setText("Clase: A");

lPublic.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lPublic.setText("IP Publica");

lIP.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N

lIP.setText("IP: 000.000.000.000");

lMaskBits.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lMaskBits.setText("Máscara de 32bits");

lIPRed.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lIPRed.setText("Ip de Red: 000.000.000.000");

lIPBroad.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lIPBroad.setText("Ip de Broadcast: 000.000.000.000");

lCantRed.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lCantRed.setText("Cantidad de Redes: 25");

lIpTipo.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lIpTipo.setText("Ip de Host");

lCantSubRed.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lCantSubRed.setText("Cantidad de Sub Redes: 25");

lCantSubRed.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

lCantSubRed.addMouseListener(new java.awt.event.MouseAdapter() {

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

lCantSubRedMouseClicked(evt);

}

});

lCantHost.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lCantHost.setText("Cantidad de Host por Red: 25");

lCantHost.setCursor(new java.awt.Cursor(java.awt.Cursor.HAND\_CURSOR));

lCantHost.addMouseListener(new java.awt.event.MouseAdapter() {

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

lCantHostMouseClicked(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(lMask)

.addComponent(lClase)

.addComponent(lPublic)

.addComponent(lIP)

.addComponent(lMaskBits)

.addComponent(lIPRed)

.addComponent(lIPBroad)

.addComponent(lCantRed)

.addComponent(lIpTipo)

.addComponent(lCantSubRed)

.addComponent(lCantHost))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(lIP)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lClase)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lPublic)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lIpTipo)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lMask)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lMaskBits)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lIPRed)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lIPBroad)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lCantRed)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lCantSubRed)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(lCantHost)

.addContainerGap())

);

pack();

}// </editor-fold>

private void llenarList(){

list=new DefaultListModel();

int top=(int)Math.pow(2, subRed);

for (int i=0;i<top;i++){

String smask=Integer.toBinaryString(i);

for (int j=0;smask.length()<subRed;j++){

smask="0"+smask;

}

if(tipo==0){

for (int j=0; j<24-subRed;j++){

smask+="0";

}

String[] sip=new String[3];

sip[0]="";

sip[1]="";

sip[2]="";

for(int j=0;j<8;j++){

sip[0]+=smask.charAt(j);

sip[1]+=smask.charAt(j+8);

sip[2]+=smask.charAt(j+16);

}

list.addElement(vec[0]+"."+Integer.parseInt(sip[0], 2)+"."+Integer.parseInt(sip[1], 2)+"."+Integer.parseInt(sip[2], 2));

}

if(tipo==1){

for (int j=0; j<16-subRed;j++){

smask+="0";

}

String[] sip=new String[2];

sip[0]="";

sip[1]="";

for(int j=0;j<8;j++){

sip[0]+=smask.charAt(j);

sip[1]+=smask.charAt(j+8);

}

list.addElement(vec[0]+"."+vec[1]+"."+Integer.parseInt(sip[0], 2)+"."+Integer.parseInt(sip[1], 2));

}

if(tipo==2){

for (int j=0; j<8-subRed;j++){

smask+="0";

}

list.addElement(vec[0]+"."+vec[1]+"."+vec[2]+"."+Integer.parseInt(smask, 2));

}

}

}

private void lCantSubRedMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

if(sn==null){

if(list==null){

llenarList();

}

sn=new frmSubNet(list,vec1);

}

sn.setLocationRelativeTo(this);

sn.setVisible(true);

}

private void lCantHostMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

if(h==null){

if(list==null){

llenarList();

}

h=new frmHost(list,vec1,bits,(int)(Math.pow(2, 32-bits)-2));

}

h.setLocationRelativeTo(this);

h.setVisible(true);

}

//-------------------------------------------------------------------------------------------------------------------------------

public class FrmPrin extends javax.swing.JFrame {

/\*\*

\* Creates new form FrmPrin

\*/

FrmGenerarIP GI;

FrmGenerarMask GM;

public FrmPrin() {

initComponents();

GI=new FrmGenerarIP(this,v1, v2, v3, v4,vb1, vb2, vb3, vb4);

GM=new FrmGenerarMask(m1, m2, m3, m4,mb1, mb2, mb3, mb4);

}

/\*\*

\* 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">

private void initComponents() {

TIp = new javax.swing.JTabbedPane();

IpDec = new javax.swing.JPanel();

v4 = new javax.swing.JTextField();

v2 = new javax.swing.JTextField();

v3 = new javax.swing.JTextField();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

v1 = new javax.swing.JTextField();

jLabel8 = new javax.swing.JLabel();

IPBin = new javax.swing.JPanel();

vb2 = new javax.swing.JTextField();

vb3 = new javax.swing.JTextField();

vb4 = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

vb1 = new javax.swing.JTextField();

TMk = new javax.swing.JTabbedPane();

MkDec = new javax.swing.JPanel();

m4 = new javax.swing.JTextField();

m2 = new javax.swing.JTextField();

m3 = new javax.swing.JTextField();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

m1 = new javax.swing.JTextField();

jLabel11 = new javax.swing.JLabel();

MkBin = new javax.swing.JPanel();

mb2 = new javax.swing.JTextField();

mb3 = new javax.swing.JTextField();

mb4 = new javax.swing.JTextField();

jLabel12 = new javax.swing.JLabel();

jLabel13 = new javax.swing.JLabel();

jLabel14 = new javax.swing.JLabel();

mb1 = new javax.swing.JTextField();

BLimpiar = new javax.swing.JButton();

BOk = new javax.swing.JButton();

BGenerar = new javax.swing.JButton();

BGenerarM = new javax.swing.JButton();

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

setResizable(false);

addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

formComponentShown(evt);

}

});

IpDec.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

IpDecComponentShown(evt);

}

});

v4.setName("3"); // NOI18N

v2.setName("1"); // NOI18N

v3.setName("2"); // NOI18N

jLabel3.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel3.setText(".");

jLabel4.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel4.setText(".");

jLabel8.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel8.setText(".");

javax.swing.GroupLayout IpDecLayout = new javax.swing.GroupLayout(IpDec);

IpDec.setLayout(IpDecLayout);

IpDecLayout.setHorizontalGroup(

IpDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(IpDecLayout.createSequentialGroup()

.addGap(98, 98, 98)

.addComponent(v1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel8)

.addGap(0, 0, 0)

.addComponent(v2, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel3)

.addGap(0, 0, 0)

.addComponent(v3, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel4)

.addGap(0, 0, 0)

.addComponent(v4, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(102, Short.MAX\_VALUE))

);

IpDecLayout.setVerticalGroup(

IpDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(IpDecLayout.createSequentialGroup()

.addContainerGap()

.addGroup(IpDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(v3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(v2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(v4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(v1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel8))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

v4.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionDec(evt);

}

});

v4.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

ipField.check(evt);

}

});

v2.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionDec(evt);

}

});

v2.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

ipField.check(evt);

}

});

v3.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionDec(evt);

}

});

v3.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

ipField.check(evt);

}

});

v1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionDec(evt);

}

});

v1.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

ipField.check(evt);

ipField.GuardarDec(v1, v2, v3, v4);

ipField.setMask(v1, m1, m2, m3, m4, mb1, mb2, mb3, mb4, TMk,BGenerarM);

}

});

TIp.addTab("IP Decimal", IpDec);

IPBin.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

IPBinComponentShown(evt);

}

});

jLabel5.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel5.setText(".");

jLabel6.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel6.setText(".");

jLabel7.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel7.setText(".");

javax.swing.GroupLayout IPBinLayout = new javax.swing.GroupLayout(IPBin);

IPBin.setLayout(IPBinLayout);

IPBinLayout.setHorizontalGroup(

IPBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(IPBinLayout.createSequentialGroup()

.addContainerGap(14, Short.MAX\_VALUE)

.addComponent(vb1, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel5)

.addGap(0, 0, 0)

.addComponent(vb2, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel6)

.addGap(0, 0, 0)

.addComponent(vb3, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel7)

.addGap(0, 0, 0)

.addComponent(vb4, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

IPBinLayout.setVerticalGroup(

IPBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(IPBinLayout.createSequentialGroup()

.addContainerGap()

.addGroup(IPBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel5)

.addComponent(jLabel6)

.addComponent(vb1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel7)

.addComponent(vb2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(vb3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(vb4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

vb2.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionBin(evt);

}

});

vb3.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionBin(evt);

}

});

vb4.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionBin(evt);

}

});

vb1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

ipField.RestricionBin(evt);

}

});

vb1.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

ipField.GuardarBin(vb1, vb2, vb3, vb4);

ipField.setMask(vb1, m1, m2, m3, m4, mb1, mb2, mb3, mb4, TMk,BGenerarM);

}

});

TIp.addTab("IP Binario", IPBin);

TMk.setEnabled(false);

MkDec.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

MkDecComponentShown(evt);

}

});

m4.setEditable(false);

m4.setName("3"); // NOI18N

m2.setEditable(false);

m2.setName("1"); // NOI18N

m3.setEditable(false);

m3.setName("2"); // NOI18N

jLabel9.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel9.setText(".");

jLabel10.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel10.setText(".");

m1.setEditable(false);

jLabel11.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel11.setText(".");

javax.swing.GroupLayout MkDecLayout = new javax.swing.GroupLayout(MkDec);

MkDec.setLayout(MkDecLayout);

MkDecLayout.setHorizontalGroup(

MkDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MkDecLayout.createSequentialGroup()

.addGap(98, 98, 98)

.addComponent(m1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel11)

.addGap(0, 0, 0)

.addComponent(m2, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel9)

.addGap(0, 0, 0)

.addComponent(m3, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel10)

.addGap(0, 0, 0)

.addComponent(m4, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(102, Short.MAX\_VALUE))

);

MkDecLayout.setVerticalGroup(

MkDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MkDecLayout.createSequentialGroup()

.addContainerGap()

.addGroup(MkDecLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(m3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(m2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel9)

.addComponent(jLabel10)

.addComponent(m4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(m1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel11))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

m4.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionDec(evt);

}

});

m4.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

maskField.check(evt);

}

});

m2.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionDec(evt);

}

});

m2.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

maskField.check(evt);

}

});

m3.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionDec(evt);

}

});

m3.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

maskField.check(evt);

}

});

m1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionDec(evt);

}

});

m1.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

maskField.check(evt);

}

});

TMk.addTab("Máscara Decimal", MkDec);

MkBin.addComponentListener(new java.awt.event.ComponentAdapter() {

public void componentShown(java.awt.event.ComponentEvent evt) {

MkBinComponentShown(evt);

}

});

mb2.setEditable(false);

mb3.setEditable(false);

jLabel12.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel12.setText(".");

jLabel13.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel13.setText(".");

jLabel14.setFont(new java.awt.Font("DejaVu Sans", 0, 14)); // NOI18N

jLabel14.setText(".");

mb1.setEditable(false);

javax.swing.GroupLayout MkBinLayout = new javax.swing.GroupLayout(MkBin);

MkBin.setLayout(MkBinLayout);

MkBinLayout.setHorizontalGroup(

MkBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MkBinLayout.createSequentialGroup()

.addContainerGap(14, Short.MAX\_VALUE)

.addComponent(mb1, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel12)

.addGap(0, 0, 0)

.addComponent(mb2, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel13)

.addGap(0, 0, 0)

.addComponent(mb3, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, 0)

.addComponent(jLabel14)

.addGap(0, 0, 0)

.addComponent(mb4, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

MkBinLayout.setVerticalGroup(

MkBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(MkBinLayout.createSequentialGroup()

.addContainerGap()

.addGroup(MkBinLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel12)

.addComponent(jLabel13)

.addComponent(mb1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel14)

.addComponent(mb2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(mb3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(mb4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

mb2.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionBin(evt);

}

});

mb3.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionBin(evt);

}

});

mb4.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionBin(evt);

}

});

mb1.addKeyListener(new java.awt.event.KeyAdapter() {

public void keyTyped(java.awt.event.KeyEvent evt) {

maskField.RestricionBin(evt);

}

});

TMk.addTab("Máscara Binario", MkBin);

BLimpiar.setText("Limpiar");

BLimpiar.addActionListener(new java.awt.event.ActionListener() {

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

BLimpiarActionPerformed(evt);

}

});

BOk.setText("OK");

BOk.setToolTipText("");

BOk.addActionListener(new java.awt.event.ActionListener() {

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

BOkActionPerformed(evt);

}

});

BGenerar.setText("Generar Ip");

BGenerar.addActionListener(new java.awt.event.ActionListener() {

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

BGenerarActionPerformed(evt);

}

});

BGenerarM.setText("Generar Máscara");

BGenerarM.setEnabled(false);

BGenerarM.addActionListener(new java.awt.event.ActionListener() {

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

BGenerarMActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(TIp, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(TMk, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(BGenerarM)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(BGenerar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(BLimpiar)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(BOk)))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap()

.addComponent(TIp, javax.swing.GroupLayout.PREFERRED\_SIZE, 69, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(TMk, javax.swing.GroupLayout.PREFERRED\_SIZE, 69, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(BLimpiar)

.addComponent(BOk)

.addComponent(BGenerar)

.addComponent(BGenerarM))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void IpDecComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

ipField.GuardarBin(vb1, vb2, vb3, vb4);

ipField.abrirDec(v1, v2, v3, v4);

}

public void setMask(){

ipField.setMask(v1, m1, m2, m3, m4, mb1, mb2, mb3, mb4, TMk,BGenerarM);

}

private void formComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

ipField.Reset();

maskField.Reset();

}

private void IPBinComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

ipField.GuardarDec(v1, v2, v3, v4);

ipField.abrirBin(vb1, vb2, vb3, vb4);

}

private void MkDecComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

maskField.GuardarBin(mb1, mb2, mb3, mb4);

maskField.abrirDec(m1, m2, m3, m4);

}

private void MkBinComponentShown(java.awt.event.ComponentEvent evt) {

// TODO add your handling code here:

maskField.GuardarDec(m1, m2, m3, m4);

maskField.abrirBin(mb1, mb2, mb3, mb4);

}

private void BLimpiarActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

ipField.Reset();

maskField.Reset();

ipField.abrirBin(vb1, vb2, vb3, vb4);

ipField.abrirDec(v1, v2, v3, v4);

maskField.abrirBin(mb1, mb2, mb3, mb4);

maskField.abrirDec(m1, m2, m3, m4);

m2.setEditable(false);

m3.setEditable(false);

m4.setEditable(false);

mb2.setEditable(false);

mb3.setEditable(false);

mb4.setEditable(false);

TMk.enable(false);

BGenerarM.setEnabled(false);

}

private void BOkActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(TIp.getSelectedIndex()==0){

ipField.GuardarDec(v1, v2, v3, v4);

}

else{

ipField.GuardarBin(vb1, vb2, vb3, vb4);

}

if(TMk.getSelectedIndex()==0){

maskField.GuardarDec(m1, m2, m3, m4);

}

else{

maskField.GuardarBin(mb1, mb2, mb3, mb4);

}

for(int i=0;i<4;i++){

if(ipField.ip[i]==-1){

ipField.ip[i]=0;

}

if(maskField.mask[i]==-1){

maskField.mask[i]=0;

}

}

ipField.abrirBin(vb1, vb2, vb3, vb4);

ipField.abrirDec(v1, v2, v3, v4);

maskField.abrirBin(mb1, mb2, mb3, mb4);

maskField.abrirDec(m1, m2, m3, m4);

if(ipField.ip[0]<=223){

if(maskField.isOK()){

FrmInfo info=new FrmInfo();

info.setLocationRelativeTo(null);

info.setVisible(true);

}

else{

JOptionPane.showMessageDialog(null, "Escribió una máscara no válida. "

+ "La máscara tiene que ser contigua. "

+ "Escriba una máscara válida.",

"Máscara incorrecta", 0);

}

}

else{

//info.setVisible(true);

FrmInfo info=new FrmInfo();

info.setLocationRelativeTo(null);

info.setVisible(true);

}

//new FrmInfo().setVisible(true);

}

private void BGenerarActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

GI.setLocationRelativeTo(this);

GI.setVisible(true);

}

private void BGenerarMActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

if(TIp.getSelectedIndex()==0){

ipField.GuardarDec(v1, v2, v3, v4);

}

else{

ipField.GuardarBin(vb1, vb2, vb3, vb4);

}

if(TMk.getSelectedIndex()==0){

maskField.GuardarDec(m1, m2, m3, m4);

}

else{

maskField.GuardarBin(mb1, mb2, mb3, mb4);

}

GM.setLocationRelativeTo(this);

GM.setVisible(true);

}

//--------------------------------------------------------------------------------------------------------------------------------

public class frmHost extends javax.swing.JFrame {

/\*\*

\* Creates new form frmHost

\*/

DefaultListModel x;

int[] vec1;

int bits;

int cant;

int ult;

DefaultListModel list;

public frmHost(DefaultListModel x,int [] vec1,int bits,int cant) {

this.x=x;

initComponents();

subred.setModel(x);

this.vec1=vec1;

this.bits=bits;

this.cant=cant;

ult=-1;

}

/\*\*

\* 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">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

subred = new javax.swing.JList();

jScrollPane2 = new javax.swing.JScrollPane();

host = new javax.swing.JList();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

subred.setModel(new javax.swing.AbstractListModel() {

String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };

public int getSize() { return strings.length; }

public Object getElementAt(int i) { return strings[i]; }

});

subred.addMouseListener(new java.awt.event.MouseAdapter() {

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

subredMouseClicked(evt);

}

});

jScrollPane1.setViewportView(subred);

host.addMouseListener(new java.awt.event.MouseAdapter() {

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

hostMouseClicked(evt);

}

});

jScrollPane2.setViewportView(host);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 144, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 250, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 300, Short.MAX\_VALUE)

.addComponent(jScrollPane2)

);

pack();

}// </editor-fold>

private int[] getIP(){

String ip = (String) x.elementAt(subred.getSelectedIndex());

int p=0;

String []vip=new String[4];

vip[0]="";

vip[1]="";

vip[2]="";

vip[3]="";

for(int i=0;i<ip.length();i++){

if(ip.charAt(i)=='.'){

p++;

}else{

vip[p]+=ip.charAt(i);

}

}

int [] t=new int[4];

t[0]=Integer.valueOf(vip[0]);

t[1]=Integer.valueOf(vip[1]);

t[2]=Integer.valueOf(vip[2]);

t[3]=Integer.valueOf(vip[3]);

return t;

}

private void subredMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

list=new DefaultListModel();

int []ipRed=getIP();

String ipFullStrng= ipField.IntToBinaryStrg(ipRed[0])+

ipField.IntToBinaryStrg(ipRed[1])+

ipField.IntToBinaryStrg(ipRed[2])+

ipField.IntToBinaryStrg(ipRed[3]);

String ipStrg="";

for (int i=0;i<bits;i++){

ipStrg+=ipFullStrng.charAt(i);

}

for (int i=1;i<cant+1;i++){

String sip=Integer.toBinaryString(i);

for (int j=0;sip.length()<32-bits;j++){

sip="0"+sip;

}

String ipFinal=ipStrg+sip;

String[] vip=new String[4];

vip[0]="";

vip[1]="";

vip[2]="";

vip[3]="";

for (int j=0;j<8;j++){

vip[0]+=ipFinal.charAt(j);

vip[1]+=ipFinal.charAt(j+8);

vip[2]+=ipFinal.charAt(j+16);

vip[3]+=ipFinal.charAt(j+24);

}

list.addElement(Integer.parseInt(vip[0], 2)+"."+Integer.parseInt(vip[1], 2)+"."+Integer.parseInt(vip[2], 2)+"."+Integer.parseInt(vip[3], 2));

}

host.setModel(list);

}

private void hostMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

String ip = (String) list.elementAt(host.getSelectedIndex());

if(ult==host.getSelectedIndex()){

int p=0;

String []vip=new String[4];

vip[0]="";

vip[1]="";

vip[2]="";

vip[3]="";

for(int i=0;i<ip.length();i++){

if(ip.charAt(i)=='.'){

p++;

}else{

vip[p]+=ip.charAt(i);

}

}

int [] t=new int[4];

t[0]=Integer.valueOf(vip[0]);

t[1]=Integer.valueOf(vip[1]);

t[2]=Integer.valueOf(vip[2]);

t[3]=Integer.valueOf(vip[3]);

ipField.ip=t.clone();

maskField.mask=vec1.clone();

FrmInfo obj=new FrmInfo();

obj.setLocationRelativeTo(this);

obj.setVisible(true);

}

else{

ult=host.getSelectedIndex();

}

}

//-------------------------------------------------------------------------------------------------------------------------------

public class frmSubNet extends javax.swing.JFrame {

/\*\*

\* Creates new form frmSubNet

\*/

int [] vec1;

DefaultListModel x;

int ult;

public frmSubNet(DefaultListModel x,int [] vec1) {

initComponents();

this.x=x;

SubNet.setModel(x);

ult=-1;

this.vec1=vec1;

}

/\*\*

\* 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">

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

SubNet = new javax.swing.JList();

setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE\_ON\_CLOSE);

setTitle("Sub Redes");

SubNet.setModel(new javax.swing.AbstractListModel() {

String[] strings = { "Item 1", "Item 2", "Item 3", "Item 4", "Item 5" };

public int getSize() { return strings.length; }

public Object getElementAt(int i) { return strings[i]; }

});

SubNet.addMouseListener(new java.awt.event.MouseAdapter() {

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

SubNetMouseClicked(evt);

}

});

jScrollPane1.setViewportView(SubNet);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 200, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 253, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void SubNetMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

String ip = (String) x.elementAt(SubNet.getSelectedIndex());

if(ult==SubNet.getSelectedIndex()){

int p=0;

String []vip=new String[4];

vip[0]="";

vip[1]="";

vip[2]="";

vip[3]="";

for(int i=0;i<ip.length();i++){

if(ip.charAt(i)=='.'){

p++;

}else{

vip[p]+=ip.charAt(i);

}

}

int [] t=new int[4];

t[0]=Integer.valueOf(vip[0]);

t[1]=Integer.valueOf(vip[1]);

t[2]=Integer.valueOf(vip[2]);

t[3]=Integer.valueOf(vip[3]);

ipField.ip=t.clone();

maskField.mask=vec1.clone();

FrmInfo obj=new FrmInfo();

obj.setLocationRelativeTo(this);

obj.setVisible(true);

}

else{

ult=SubNet.getSelectedIndex();

}

}