

## ANEXO 2. CODIGO CALCULADORA CIDR

```
package calculadoracidr;
import java.util.regex.*;
import java.util.StringTokenizer;
/**
 *
 * @author AB
 */
public class Calculos {
    public int red;
    public String tipo;
    public String valdat,dat;
    public int []segred=new int [3];

    public String CalculoRed(int red)
    {
        //Se valida la variable red (valor del primer Textfiel de la dirección IP) para
        //saber a que clase corresponde.

        if (red>=0 && red<=127){
            tipo="A";
        }
        else{

            if (red>=128 && red<=191){
                tipo="B";
            }
            else
            {
                if(red>=192 && red<=223){
                    tipo="C";
                }
                else{
                    tipo="NE";
                }
            }
        }
        return tipo; // dependiendo del rango de valores, se regresa un string con la letra de la clase
    }

    public boolean Validacion(String valdat){
        boolean resulval=true;

        Pattern p = Pattern.compile("[0-9]"); //Se toma la una expresión regular [0-9]para restringir solo numeros.
        Matcher m = p.matcher(valdat); //Se hace la comparacion con el dato.
        if (!m.find()){ //Si el valor del dato es diferente a un número la variable boleana
            resulval=false; //es igual a falso.
        }

        return resulval;
    }

    public boolean ValidacionLim(String numero){

        int numerol;
        numerol=Integer.parseInt(numero); //Se hace la conversión de String a integer para evaluar el dato.
        boolean validar=true;

        if (numerol>255){ //Se verifica el número que no sea mayor a 255. Esto es para el caso
            validar=false; //de las direcciones.
        }

        return validar;
    }

    public boolean ValidacionLimNet(String numero){

        int numerol;
        numerol=Integer.parseInt(numero); //Conversión de string a integer.
        boolean validar=true;

        if (numerol>31){ //Se verifica el número que no sea mayor a 31. Esto es para el caso de Netmask.
            validar=false;
        }

        return validar;
    }

    public String[] Separacion(String dat){

        StringTokenizer tokens = new StringTokenizer(dat,"."); //Para poder realizar la separación de la mascara en
        //4 fragmentos. Se lee la cadena y cuando encuentre
        //un punto, lo separa del resto

        int numdat=tokens.countTokens();
        int k=0;
        Calculos a=new Calculos();
        String []ip=new String [numdat];

        while(tokens.hasMoreTokens()){ //Se lee toda la cadena de la mascara,se separa por tokens y se
            ip[k]=tokens.nextToken(); //almacena en la variable ip
            k++;
        }

        return ip;
    }

    public String Calculo(String [] sep, int []red){

        int []resred=new int[4];
        String []resredl=new String[4];
        String resred2;

        resred[0]=(Integer.parseInt(sep[0]))& red[0];//Se recibe el octeto separado de la mascara (en este caso 1er octeto)
        resred[1]=(Integer.parseInt(sep[1]))& red[1];//y se realiza una AND con el octeto de la red (1er octeto). El
        resred[2]=(Integer.parseInt(sep[2]))& red[2];//resultado se almacena en el arrglo resred.
        resred[3]=(Integer.parseInt(sep[3]))& red[3];

        //Conversion de String a Integer de resred, para su manipulacion como número
        resredl[0]=Integer.toString(resred[0]);
        resredl[1]=Integer.toString(resred[1]);
        resredl[2]=Integer.toString(resred[2]);
        resredl[3]=Integer.toString(resred[3]);

        //Se forma la cadena del resultado de obtenido con la AND para imprimirlo en pantalla
        resred2=resred[0]+"."+resred[1]+"."+resred[2]+"."+resred[3];

        return resred2;
    }

    public String HostMin(String [] sep, int []red){

        int []resred=new int[4];
```

## ANEXO 2. CODIGO CALCULADORA CIDR

```
String []resred1=new String[4];
String resred2;

resred[0]=(Integer.parseInt(sep[0]))& red[0];// Se realiza la misma operación para calcular el host minimo
resred[1]=(Integer.parseInt(sep[1]))& red[1];//del rango de red
resred[2]=(Integer.parseInt(sep[2]))& red[2];
resred[3]=(Integer.parseInt(sep[3]))& red[3];

resred1[0]=Integer.toString(resred[0]);//Conversion de String a Integer de resred, para su manipulacion como número
resred1[1]=Integer.toString(resred[1]);
resred1[2]=Integer.toString(resred[2]);
resred1[3]=Integer.toString(resred[3]);

//Se forma la cadena para imprimir en pantalla la dirección del host minimo, donde se suma "1" al ultimo octeto
resred2=resred[0]+"."+resred[1]+"."+resred[2]+"."+resred[3]+1;

return resred2;
}

public int [] Calculo1(String [] sep, int []red){
int []resred=new int[4];
String []resred1=new String[4];

//Conversion de String a Integer de resred, para su manipulacion como número
resred[0]=(Integer.parseInt(sep[0]))& red[0];
resred[1]=(Integer.parseInt(sep[1]))& red[1];
resred[2]=(Integer.parseInt(sep[2]))& red[2];
resred[3]=(Integer.parseInt(sep[3]))& red[3];

return resred;
}

public int [] Wilcard(String [] sep1){

int []wild=new int [4];
String wild2;
//Para el calculo de la Wildcard, se toma la mascara separada (convirtiendo el dato en entero) y se resta 255
wild[0]=255-(Integer.parseInt(sep1[0]));
wild[1]=255-(Integer.parseInt(sep1[1]));
wild[2]=255-(Integer.parseInt(sep1[2]));
wild[3]=255-(Integer.parseInt(sep1[3]));

//Se forma la cadena del resultado de obtenido con la AND para imprimirlo en pantalla

wild2=wild[0]+"."+wild[1]+"."+wild[2]+"."+wild[3];

return wild;
}

public String Wilcard1(String [] sep1){

int []wild=new int [4];
String wild2;
//Para el calculo de la Wildcard, se toma la mascara separada (convirtiendo el dato en entero) y se resta 255
wild[0]=255-(Integer.parseInt(sep1[0]));
wild[1]=255-(Integer.parseInt(sep1[1]));
wild[2]=255-(Integer.parseInt(sep1[2]));
wild[3]=255-(Integer.parseInt(sep1[3]));

//Se forma la cadena del resultado de obtenido con la AND para imprimirlo en pantalla

wild2=wild[0]+"."+wild[1]+"."+wild[2]+"."+wild[3];

return wild2;
}

public String Maxrango (int [] w, int [] r ){

int []maxrango=new int[4];
String maxrango2;

//Para el calculo del rango, se realiza la suma de la wilcard(w) y la red (r), dando como resultado la dirección
//del ultimo host que se obtendra de la red
maxrango[0]=w[0]+r[0];
maxrango[1]=w[1]+r[1];
maxrango[2]=w[2]+r[2];
maxrango[3]=w[3]+r[3];

maxrango2=maxrango[0]+"."+maxrango[1]+"."+maxrango[2]+"."+maxrango[3];

return maxrango2;
}

public String HostMax (int [] w, int [] r ){

int []maxrango=new int[4];
String maxrango2;

//Para el calculo del rango, se realiza la suma de la wilcard(w) y la red (r), dando como resultado la dirección
//del ultimo host que se obtendra de la red
maxrango[0]=w[0]+r[0];
maxrango[1]=w[1]+r[1];
maxrango[2]=w[2]+r[2];
maxrango[3]=w[3]+r[3];

//Se forma la cadena para imprimir en pantalla la dirección del host maximo, donde se resta "1" al ultimo octeto
maxrango2=maxrango[0]+"."+maxrango[1]+"."+maxrango[2]+"."+maxrango[3]-1;

return maxrango2;
}

//NOTA: Se tienen metodos parecidos con nombre distinto, como es el caso de Caluculo y Calculo1, esto es por el
//tipo de dato que regresa. En uno es String y en el otro es Integer o entero. Y lo maneje de esta forma, por
//simplicidad y para el manejo de la información.
}
```

## ANEXO 2. CODIGO CALCULADORA CIDR

```
/*
 * CalCIDR.java
 *
 * Created on 31/05/2012, 08:53:29 PM
 */

package calculadoracidr;

import java.awt.*;
import javax.swing.*;

/**
 * @author ABRAHAM BROWDER SERRANO
 */
public class CalCIDR extends javax.swing.JFrame {

    /** Creates new form CalCIDR */
    public CalCIDR() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // Generated Code

    private void ComBoxMascActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int posicion;
        int []red=new int[4];
        String []dato=new String [4];
        Calculos a=new Calculos();
        posicion=ComBoxMasc.getSelectedIndex();

        //Se toma el dato ingresado en la caja de texto de la "dirección Ip del bloque CIDR" y se almacena
        //en la variable dato.
        dato[0]=TxtRed.getText();
        dato[1]=TxtRed1.getText();
        dato[2]=TxtRed2.getText();
        dato[3]=TxtRed3.getText();
        //Se hace una conversión de String a Integer.
        red[0] = Integer.parseInt(dato[0]);
        red[1] = Integer.parseInt(dato[1]);
        red[2] = Integer.parseInt(dato[2]);
        red[3] = Integer.parseInt(dato[3]);

        switch (posicion) {
            case 0:
                //De la posición del dato en el ComboBox que corresponde a la máscara, se le asigna un valor
                //a la caja de texto Netmask.
                TxtNetmask.setText("1");
                //De la posición del dato en el ComboBox que corresponde a la máscara, se le asigna un valor
                //al ComboBox N° Máximo de Host.
                ComBoxMaxHost.setSelectedIndex(0);
                //Para obtener la dirección de red, hay que mandar llamar el método Separación de la clase
                //Calculos y enviar el valor de la máscara. Después se manda llamar el método Calculo, con
                //el resultado del método Separación y la red.
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red));
                //Para obtener la wildcard, se manda llamar el método Separación y el resultado se envía al
                //método Wilcard1.
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(0))););
                //Para obtener el rango, se envía la misma información para la dirección de red y se agrega
                //el resultado del método Maxrango, el cual requiere dos entradas. La primera esta compuesta
                //por el resultado del método Wilcard y el segundo, por el resultado del método Calculo.
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(0))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red));
                //Para obtener la notación CIDR, se requiere del resultado del método Calculo y el dato ingresado en la
                //caja de texto Netmask.
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red))+ "/" + TxtNetmask.getText());
                //Para obtener el broadcast, se toma el resultado obtenido en el método Maxrango.
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(0))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red));
                //Para obtener el host mínimo, se toma el resultado del método HostMin, el cual requiere al método
                //Separación y el valor de red.
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(0)),red));
                //Para obtener el host máximo, se toma el resultado del método HostMax, el cual requiere al método
                //Wilcard y Calculo1.
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(0))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(0)),red));
            break;
            case 1:
                TxtNetmask.setText("2");
                ComBoxMaxHost.setSelectedIndex(1);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(1))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(1))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(1))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(1)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(1))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(1)),red));
            break;
            case 2:
                TxtNetmask.setText("3");
                ComBoxMaxHost.setSelectedIndex(2);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(2))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(2))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(2))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(2)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(2))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(2)),red));
            break;
            case 3:
                TxtNetmask.setText("4");
                ComBoxMaxHost.setSelectedIndex(3);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(3))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(3))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(3))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(3)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(3))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(3)),red));
            break;
            case 4:
                TxtNetmask.setText("5");
                ComBoxMaxHost.setSelectedIndex(4);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(4))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(4))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(4))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(4)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(4))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(4)),red));
            break;
            case 5:
                TxtNetmask.setText("6");
                ComBoxMaxHost.setSelectedIndex(5);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(5))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(5))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(5))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(5)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(5))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(5)),red));
            break;
            case 6:
                TxtNetmask.setText("7");
                ComBoxMaxHost.setSelectedIndex(6);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(6))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(6))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(6))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(6)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(6))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(6)),red));
            break;
            case 7:
                TxtNetmask.setText("8");
                ComBoxMaxHost.setSelectedIndex(7);
                TxtDirRed.setText(a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red));
                TxtWild.setText(a.Wilcard1(a.Separacion((String) ComBoxMasc.getItemAt(7))););
                TxtRango.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red))+ " - "+a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(7))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red));
                TxtNotacion.setText((a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red))+ "/" + TxtNetmask.getText());
                TxtBroad.setText(a.Maxrango(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(7))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red));
                TxtHMin.setText(a.HostMin(a.Separacion((String) ComBoxMasc.getItemAt(7)),red));
                TxtHMax.setText(a.HostMax(a.Wilcard(a.Separacion((String) ComBoxMasc.getItemAt(7))),a.Calculo(a.Separacion((String) ComBoxMasc.getItemAt(7)),red));
            break;
            case 8:
                TxtNetmask.setText("9");
        }
    }
}
```







## ANEXO 2. CODIGO CALCULADORA CIDR

```
ComBoxMasc.setSelectedIndex(10);
ComBoxMaxHost.setSelectedIndex(10);
break;
case 12:
ComBoxMasc.setSelectedIndex(11);
ComBoxMaxHost.setSelectedIndex(11);
break;
case 13:
ComBoxMasc.setSelectedIndex(12);
ComBoxMaxHost.setSelectedIndex(12);
break;
case 14:
ComBoxMasc.setSelectedIndex(13);
ComBoxMaxHost.setSelectedIndex(13);
break;
case 15:
ComBoxMasc.setSelectedIndex(14);
ComBoxMaxHost.setSelectedIndex(14);
break;
case 16:
ComBoxMasc.setSelectedIndex(15);
ComBoxMaxHost.setSelectedIndex(15);
break;
case 17:
ComBoxMasc.setSelectedIndex(16);
ComBoxMaxHost.setSelectedIndex(16);
break;
case 18:
ComBoxMasc.setSelectedIndex(17);
ComBoxMaxHost.setSelectedIndex(17);
break;
case 19:
ComBoxMasc.setSelectedIndex(18);
ComBoxMaxHost.setSelectedIndex(18);
break;
case 20:
ComBoxMasc.setSelectedIndex(19);
ComBoxMaxHost.setSelectedIndex(19);
break;
case 21:
ComBoxMasc.setSelectedIndex(20);
ComBoxMaxHost.setSelectedIndex(20);
break;
case 22:
ComBoxMasc.setSelectedIndex(21);
ComBoxMaxHost.setSelectedIndex(21);
break;
case 23:
ComBoxMasc.setSelectedIndex(22);
ComBoxMaxHost.setSelectedIndex(22);
break;
case 24:
ComBoxMasc.setSelectedIndex(23);
ComBoxMaxHost.setSelectedIndex(23);
break;
case 25:
ComBoxMasc.setSelectedIndex(24);
ComBoxMaxHost.setSelectedIndex(24);
break;
case 26:
ComBoxMasc.setSelectedIndex(25);
ComBoxMaxHost.setSelectedIndex(25);
break;
case 27:
ComBoxMasc.setSelectedIndex(26);
ComBoxMaxHost.setSelectedIndex(26);
break;
case 28:
ComBoxMasc.setSelectedIndex(27);
ComBoxMaxHost.setSelectedIndex(27);
break;
case 29:
ComBoxMasc.setSelectedIndex(28);
ComBoxMaxHost.setSelectedIndex(28);
break;
case 30:
ComBoxMasc.setSelectedIndex(29);
ComBoxMaxHost.setSelectedIndex(29);
break;
case 31:
ComBoxMasc.setSelectedIndex(30);
ComBoxMaxHost.setSelectedIndex(30);
break;
case 32:
ComBoxMasc.setSelectedIndex(31);
ComBoxMaxHost.setSelectedIndex(31);
System.out.println("soy 31");
break;

default:
System.out.println("error");
break;
}
}

private void ComBoxMaxHostActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
int posicion;
String dato;

posicion=ComBoxMaxHost.getSelectedIndex();//Con la posición de N° Maximo de Host, se selecciona la
//la posición del Combo Box de la Mascara.

switch(posicion){
case 0:
ComBoxMasc.setSelectedIndex(0);
break;
case 1:
ComBoxMasc.setSelectedIndex(1);
break;
case 2:
ComBoxMasc.setSelectedIndex(2);
break;
case 3:
ComBoxMasc.setSelectedIndex(3);
break;
case 4:
ComBoxMasc.setSelectedIndex(4);
break;
case 5:
ComBoxMasc.setSelectedIndex(5);
break;
case 6:
ComBoxMasc.setSelectedIndex(6);
break;
case 7:
ComBoxMasc.setSelectedIndex(7);
break;
case 8:
ComBoxMasc.setSelectedIndex(8);
break;
case 9:
ComBoxMasc.setSelectedIndex(9);
break;
case 10:
ComBoxMasc.setSelectedIndex(10);
break;
case 11:
ComBoxMasc.setSelectedIndex(11);
break;
case 12:
ComBoxMasc.setSelectedIndex(12);
break;
case 13:
ComBoxMasc.setSelectedIndex(13);
break;
case 14:
ComBoxMasc.setSelectedIndex(14);
break;
case 15:
ComBoxMasc.setSelectedIndex(15);
break;
case 16:
ComBoxMasc.setSelectedIndex(16);
break;
}
```

## ANEXO 2. CODIGO CALCULADORA CIDR

```
case 17:
    ComBoxMasc.setSelectedIndex(17);
break;
case 18:
    ComBoxMasc.setSelectedIndex(18);
break;
case 19:
    ComBoxMasc.setSelectedIndex(19);
break;
case 20:
    ComBoxMasc.setSelectedIndex(20);
break;
case 21:
    ComBoxMasc.setSelectedIndex(21);
break;
case 22:
    ComBoxMasc.setSelectedIndex(22);
break;
case 23:
    ComBoxMasc.setSelectedIndex(23);
break;
case 24:
    ComBoxMasc.setSelectedIndex(24);
break;
case 25:
    ComBoxMasc.setSelectedIndex(25);
break;
case 26:
    ComBoxMasc.setSelectedIndex(26);
break;
case 27:
    ComBoxMasc.setSelectedIndex(27);
break;
case 28:
    ComBoxMasc.setSelectedIndex(28);
break;
case 29:
    ComBoxMasc.setSelectedIndex(29);
break;
case 30:
    ComBoxMasc.setSelectedIndex(30);
break;
case 31:
    ComBoxMasc.setSelectedIndex(31);
break;

default:
    break;
}
}

private void TxtWildActionPerformed (java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void TxtRedKeyPressed (java.awt.event.KeyEvent evt) {
    Calculos a=new Calculos();
    //Se valida que la caja de texto no contenga mas de tres digitos, de ser asi, se borra para volver a escribir
    if (TxtRed.getText().length()>2){
        TxtRed.setSelectionStart(0);
    }
}

private void TxtRed1KeyPressed (java.awt.event.KeyEvent evt) {
    Calculos a=new Calculos();
    //Se valida que la caja de texto no contenga mas de tres digitos, de ser asi, se borra para volver a escribir
    if (TxtRed1.getText().length()>2){
        TxtRed1.setSelectionStart(0);
    }
}

private void TxtRed2KeyPressed (java.awt.event.KeyEvent evt) {
    Calculos a=new Calculos();
    //Se valida que la caja de texto no contenga mas de tres digitos, de ser asi, se borra para volver a escribir
    if (TxtRed2.getText().length()>2){
        TxtRed2.setSelectionStart(0);
    }
}

private void TxtRed3KeyPressed (java.awt.event.KeyEvent evt) {
    Calculos a=new Calculos();
    //Se valida que la caja de texto no contenga mas de tres digitos, de ser asi, se borra para volver a escribir
    if (TxtRed3.getText().length()>2){
        TxtRed3.setSelectionStart(0);
    }
}

private void TxtRedFocusLost (java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    int red;
    String dato0;
    String numbin;
    Calculos a=new Calculos();

    TxtRed.setSelectionStart(0); // Se selecciona el texto.
    if (TxtRed.getText().length()<1){ //Se valida que la caja de texto contenga digitos, de no ser asi, se escribe un cero.
        TxtRed.setText("0");
    }

    dato0=TxtRed.getText();
    //Se llama al metodo Validacion, el cual valida que se ingrese solo numeros, de no ser asi. se escribe "0"
    if (a.Validacion(dato0)==false){
        TxtRed.setText("0");
    }
    //Se llama al metodo ValidacionLim, el cual valida que no se ingresen numeros mayor a 255, de ser asi. se escribe "255"
    if (a.ValidacionLim(TxtRed.getText())==false){
        TxtRed.setText("255");
    }
    red = Integer.parseInt(dato0);
    // Validación de tipo de red.
    TxtTipoRed.setText((String)a.CalculoRed(red));
}

private void TxtRed1FocusLost (java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    Calculos a=new Calculos();

    String dato1;
    dato1=TxtRed1.getText();

    //Se valida que la caja de texto contenga digitos, de no ser asi, se escribe un cero.
    TxtRed1.setSelectionStart(0);
    if (TxtRed1.getText().length()<1){
        TxtRed1.setText("0");
    }
    //Se llama al metodo Validacion, el cual valida que se ingrese solo numeros, de no ser asi. se escribe "0"
    if (a.Validacion(dato1)==false){
        TxtRed1.setText("0");
    }
    //Se llama al metodo ValidacionLim, el cual valida que no se ingresen numeros mayor a 255, de ser asi. se escribe "255"
    if (a.ValidacionLim(TxtRed1.getText())==false){
        TxtRed1.setText("255");
    }
}

private void TxtRed2FocusLost (java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    Calculos a=new Calculos();
    String dato2;
    dato2=TxtRed2.getText();

    TxtRed2.setSelectionStart(0);
    //Se valida que la caja de texto contenga digitos, de no ser asi, se escribe un cero.
    if (TxtRed2.getText().length()<1){
        TxtRed2.setText("0");
    }
}
```

## ANEXO 2. CODIGO CALCULADORA CIDR

```
}
//Se llama al metodo Validacion, el cual valida que se ingrese solo numeros, de no ser asi. se escribe "0"
if(a.Validacion(dato2)==false){
    TxtRed2.setText("0");
}
//Se llama al metodo ValidacionLim, el cual valida que no se ingresen numeros mayor a 255, de ser asi. se escribe "255"
if(a.ValidacionLim(TxtRed2.getText())==false){
    TxtRed2.setText("255");
}
}

private void TxtRed3FocusLost (java.awt.event.FocusEvent evt) {
    // TODO add your handling code here:
    Calculos a=new Calculos();
    String dato3;
    dato3=TxtRed3.getText();

    TxtRed3.setSelectionStart(0);
//Se valida que la caja de texto contenga digitos, de no ser asi, se escribe un cero.
if(TxtRed3.getText().length()<1){
    TxtRed3.setText("0");
}
//Se llama al metodo Validacion, el cual valida que se ingrese solo numeros, de no ser asi. se escribe "0"
if(a.Validacion(dato3)==false){
    TxtRed3.setText("0");
}
//Se llama al metodo ValidacionLim, el cual valida que no se ingresen numeros mayor a 255, de ser asi. se escribe "255"
if(a.ValidacionLim(TxtRed3.getText())==false){
    TxtRed3.setText("255");
}
}

private void TxtNetmaskKeyPressed (java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    String net;
    Calculos a=new Calculos();

    net=TxtNetmask.getText();

//Se llama al metodo Validacion, el cual valida que se ingrese solo numeros, de no ser asi. se escribe "1"
if(a.Validacion(net)==false){
    TxtNetmask.setText("1");
    TxtNetmask.setSelectionStart(0);
}
//Se llama al metodo ValidacionLim, el cual valida que no se ingresen numeros mayor a 31, de ser asi. se escribe "31"
if(a.ValidacionLimNet(TxtNetmask.getText())==false){
    TxtNetmask.setText("31");
}
//Se valida que la caja de texto contenga digitos, de no ser asi, se escribe un cero.
if(TxtNetmask.getText().length()>1){
    TxtNetmask.setSelectionStart(0);
}
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new CalCIDR().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JComboBox ComBoxMasc;
private javax.swing.JComboBox ComBoxMaxHost;
private javax.swing.JLabel LabBroat;
private javax.swing.JLabel LabHostMax;
private javax.swing.JLabel LabHostMin;
private javax.swing.JLabel LabMascaraCIDR;
private javax.swing.JLabel LabNot;
private javax.swing.JLabel LabRango;
private javax.swing.JLabel LabRed;
private javax.swing.JLabel LabWil;
private javax.swing.JLabel LabelDir;
private javax.swing.JLabel LblPunto;
private javax.swing.JLabel LblPunto1;
private javax.swing.JLabel LblPunto2;
private javax.swing.JLabel LblTipoRed;
private javax.swing.JTextField TxtBroad;
private javax.swing.JTextField TxtDirRed;
private javax.swing.JTextField TxtHMax;
private javax.swing.JTextField TxtHMin;
private javax.swing.JTextField TxtNetmask;
private javax.swing.JTextField TxtNotacion;
private javax.swing.JTextField TxtRango;
private javax.swing.JTextField TxtRed;
private javax.swing.JTextField TxtRed1;
private javax.swing.JTextField TxtRed2;
private javax.swing.JTextField TxtRed3;
private javax.swing.JTextField TxtTipoRed;
private javax.swing.JTextField TxtValIP;
private javax.swing.JTextField TxtWild;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
// End of variables declaration
}
}
```



## ANEXO 2. CODIGO SISTEMA EXPERTO

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * Ayudal.java
 *
 */

package sistemaexperto;

/**
 *
 * @author AB
 */
public class Ayudal extends javax.swing.JDialog {

    private SistExp padre=(SistExp)this.getParent();
    /** Creates new form Ayudal */
    public Ayudal(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        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">
    private void initComponents() {

        LabImal = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Ayuda. Versión de IOS Cisco");

        LabImal.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/showversion.JPG"))); // NOI18N

        jTextField1.setEditable(false);
        jTextField1.setText(" Para conocer el nombre de la imagen y la versión que está corriendo en un router Cisco antes de comenzar ");
        jTextField1.setBorder(null);

        jTextField2.setEditable(false);
        jTextField2.setText("a configurarlo, usamos el comando SHOW VERSION como se muestra en la imagen. ");
        jTextField2.setBorder(null);

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .add(layout.createSequentialGroup()
                                .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .add(layout.createSequentialGroup()
                                        .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                            .add(layout.createSequentialGroup()
                                                .addContainerGap()
                                                .addComponent(jLabel1)
                                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 600, javax.swing.GroupLayout.PREFERRED_SIZE)
                                                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, 514, javax.swing.GroupLayout.PREFERRED_SIZE))
                                                .addGroup(layout.createSequentialGroup()
                                                    .addGap(37, 37, 37)
                                                    .addComponent(LabImal, javax.swing.GroupLayout.PREFERRED_SIZE, 563, javax.swing.GroupLayout.PREFERRED_SIZE))
                                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                            .add(layout.createSequentialGroup()
                                                .addComponent(LabImal, javax.swing.GroupLayout.PREFERRED_SIZE, 563, javax.swing.GroupLayout.PREFERRED_SIZE))
                                                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                        )
                                    .add(layout.createSequentialGroup()
                                        .addComponent(LabImal, javax.swing.GroupLayout.PREFERRED_SIZE, 563, javax.swing.GroupLayout.PREFERRED_SIZE))
                                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                )
                            )
                        )
                    )
                )
            .addContainerGap());
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(layout.createSequentialGroup()
                            .add(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()
                                    .addComponent(jLabel1)
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .add(layout.createSequentialGroup()
                                            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 14, javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                                        .addGap(16, 16, 16)
                                        .addComponent(LabImal, javax.swing.GroupLayout.PREFERRED_SIZE, 80, javax.swing.GroupLayout.PREFERRED_SIZE)
                                        .addContainerGap(24, Short.MAX_VALUE))
                                    )
                                )
                            )
                        )
                    )
                )
            .addContainerGap());
        pack();
    }
    // </editor-fold>

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Ayudal dialog = new Ayudal(new javax.swing.JFrame(), true);
                dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JLabel LabImal;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}

```



## ANEXO 2. CODIGO SISTEMA EXPERTO

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * Ayuda3.java
 *
 */

package sistemaexperto;

/**
 *
 * @author AB
 */
public class Ayuda3 extends javax.swing.JDialog {

    /** Creates new form Ayuda3 */
    private SistExp padre=(SistExp)this.getParent();
    public Ayuda3(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        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">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        TxtConsejo1 = new javax.swing.JTextField();
        TxtConsejo2 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Ayuda. Protocolo de enrutamiento");

        jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N

        TxtConsejo1.setEditable(false);
        TxtConsejo1.setText("Para verificar que protocolo de enrutamiento que se utiliza, se puede observar con el comando SHOW IP.");
        TxtConsejo1.setBorder(null);

        TxtConsejo2.setEditable(false);
        TxtConsejo2.setText("Para comprobar la versión del protocolo, se utiliza el comando SHOW IP PROTOCOLS.");
        TxtConsejo2.setBorder(null);

        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/show ip1.jpg"))); // NOI18N

        jLabel3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/rip.jpg"))); // NOI18N

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel1)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(TxtConsejo2, javax.swing.GroupLayout.PREFERRED_SIZE, 502, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel2)
                            .addComponent(TxtConsejo1, javax.swing.GroupLayout.DEFAULT_SIZE, 579, Short.MAX_VALUE))
                        .addComponent(jLabel3)
                    )
                )
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel1)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(TxtConsejo1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel2)
                        .addComponent(TxtConsejo2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel3)
                    )
                )
        );

        pack();
    } // </editor-fold>

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Ayuda3 dialog = new Ayuda3(new javax.swing.JFrame(), true);
                dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                    public void windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JTextField TxtConsejo1;
    private javax.swing.JTextField TxtConsejo2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    // End of variables declaration
}

```

## ANEXO 2. CODIGO SISTEMA EXPERTO

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/*
 * SistExp.java
 *
 * Created on 4/07/2012, 03:53:35 PM
 */

package sistemaexperto;

import java.util.Locale;
import javax.swing.*;
/**
 *
 * @author AB
 */
public class SistExp extends javax.swing.JFrame {

    /** Creates new form SistExp */
    public SistExp() {
        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">
    private void initComponents() {

        LabPreg1 = new javax.swing.JLabel();
        CmbPreg1 = new javax.swing.JComboBox();
        LabPreg2 = new javax.swing.JLabel();
        CmbPreg2 = new javax.swing.JComboBox();
        CmbPreg3 = new javax.swing.JComboBox();
        LabPreg3 = new javax.swing.JLabel();
        LabPreg31 = new javax.swing.JLabel();
        LabPreg4 = new javax.swing.JLabel();
        CmbPreg4 = new javax.swing.JComboBox();
        LabPreg5 = new javax.swing.JLabel();
        CmbPreg5 = new javax.swing.JComboBox();
        LabPreg6 = new javax.swing.JLabel();
        LabPreg61 = new javax.swing.JLabel();
        CmbPreg6 = new javax.swing.JComboBox();
        ButEvaluar = new javax.swing.JButton();
        ButAyuda1 = new javax.swing.JButton();
        ButAyuda3 = new javax.swing.JButton();
        ButAyuda2 = new javax.swing.JButton();
        ButAyuda4 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Sistema Experto");

        LabPreg1.setText("1.¿Se tiene cable de consola RJ-45-DB9 y una Computadora con HyperTerminal para configurar el router?");

        CmbPreg1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));
        CmbPreg1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                CmbPreg1ActionPerformed(evt);
            }
        });

        LabPreg2.setText("2.¿Se cuenta con un router Cisco con versión de Cisco IOS 12. x T, 12.2S, 12.2SY, 12.3, 12.4, 15.0, 15.0S, 15.0SY, o 15.1?");

        CmbPreg2.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));
        CmbPreg2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                CmbPreg2ActionPerformed(evt);
            }
        });

        CmbPreg3.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));

        LabPreg3.setText("3.¿Se cuenta con Sistema Operativo en los host como: Windows XP con SP1 o superior, Windows 7, Windows Vista");

        LabPreg31.setText(" Windows Server 2003, Windows Server 2008, Linux con Kernel 2.2 o superior o Mac OS X 10.6 o superior?");

        LabPreg4.setText("4.¿Se cuenta con un servidor DHCP?");

        CmbPreg4.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));

        LabPreg5.setText("5.¿Se tiene una red clase A, B o C?");

        CmbPreg5.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));

        LabPreg6.setText("6.¿Se cuenta con protocolos de enrutamiento sin clases como: RIPv.2, EIGRP, OSPF o BGPv4 y más de una red ");

        LabPreg61.setText("contigua a la utilizada?");

        CmbPreg6.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Elegir opción", "Si", "No" }));

        ButEvaluar.setBackground(new java.awt.Color(102, 153, 255));
        ButEvaluar.setFont(new java.awt.Font("Tahoma", 1, 18));
        ButEvaluar.setForeground(new java.awt.Color(255, 255, 255));
        ButEvaluar.setText("Evaluar");
        ButEvaluar.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ButEvaluarActionPerformed(evt);
            }
        });

        ButAyuda1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N
        ButAyuda1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ButAyuda1ActionPerformed(evt);
            }
        });

        ButAyuda3.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N
        ButAyuda3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ButAyuda3ActionPerformed(evt);
            }
        });

        ButAyuda2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N
        ButAyuda2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                ButAyuda2ActionPerformed(evt);
            }
        });

        ButAyuda4.setIcon(new javax.swing.ImageIcon(getClass().getResource("/sistemaexperto/icono-ayuda.jpg"))); // NOI18N
        ButAyuda4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {

```

## ANEXO 2. CODIGO SISTEMA EXPERTO

```
        ButAyuda4ActionPerformed(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()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(ButEvaluar)
                .addComponent(CmbPreg2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(CmbPreg1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(LabPreg31, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(LabPreg3, javax.swing.GroupLayout.Alignment.LEADING))
                .addComponent(CmbPreg5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(LabPreg1, javax.swing.GroupLayout.PREFERRED_SIZE, 587, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
            .addGap(27, 27, 27)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(ButAyuda4, 0, 0, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(ButAyuda1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                    .addComponent(ButAyuda3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                    .addComponent(ButAyuda2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE))
            )
            .addGap(51, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addComponent(ButAyuda4, 0, 0, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                .addComponent(ButAyuda1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                .addComponent(ButAyuda3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                .addComponent(ButAyuda2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE))
            .addGap(51, Short.MAX_VALUE))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(ButAyuda4, 0, 0, Short.MAX_VALUE)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                        .addComponent(ButAyuda1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                        .addComponent(ButAyuda3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                        .addComponent(ButAyuda2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE))
                    .addGap(51, Short.MAX_VALUE))
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(ButEvaluar)
                    .addComponent(CmbPreg2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(CmbPreg1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                        .addComponent(LabPreg31, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 16, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(LabPreg3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(LabPreg4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(LabPreg5, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 54, Short.MAX_VALUE))
                    .addComponent(ButAyuda4, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(ButAyuda3, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(ButAyuda2, javax.swing.GroupLayout.PREFERRED_SIZE, 22, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(ButAyuda1, javax.swing.GroupLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZE)
                )
            )
            .addGap(9, 9, 9)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(ButAyuda4, 0, 0, Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(ButAyuda1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                    .addComponent(ButAyuda3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE)
                    .addComponent(ButAyuda2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 23, Short.MAX_VALUE))
                .addGap(51, Short.MAX_VALUE))
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(ButEvaluar)
                .addComponent(CmbPreg2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(CmbPreg1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                    .addComponent(LabPreg31, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 16, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(LabPreg3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(LabPreg4, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(LabPreg5, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED_SIZE, 54, Short.MAX_VALUE))
                .addComponent(ButAyuda4, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(ButAyuda3, javax.swing.GroupLayout.PREFERRED_SIZE, 24, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(ButAyuda2, javax.swing.GroupLayout.PREFERRED_SIZE, 22, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(ButAyuda1, javax.swing.GroupLayout.PREFERRED_SIZE, 26, javax.swing.GroupLayout.PREFERRED_SIZE)
            )
        )
    );
pack();
}

private void CmbPreg1ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void CmbPreg2ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void ButEvaluarActionPerformed(java.awt.event.ActionEvent evt) {
    int r1, r2, r3, r4, r5, r6;
    Mecanismo m= new Mecanismo();
    //Se asigna un numero a la variable que proviene de la opción seleccionada del combo box
    r1=CmbPreg1.getSelectedIndex();
    r2=CmbPreg2.getSelectedIndex();
    r3=CmbPreg3.getSelectedIndex();
    r4=CmbPreg4.getSelectedIndex();
    r5=CmbPreg5.getSelectedIndex();
    r6=CmbPreg6.getSelectedIndex();

    if(r1==0 || r2==0 || r3==0 || r4==0 || r5==0 || r6==0){
        //Ventana emergente de error
        JOptionPane.showMessageDialog(null, "Pregunta(s) sin contestar", "Resultado", JOptionPane.ERROR_MESSAGE);
    }
    else{
        //Ventana emergente del resultado
        JOptionPane.showMessageDialog(null, m.Metodo(r1,r2,r3,r4,r5,r6), "Resultado", JOptionPane.INFORMATION_MESSAGE);
    }
}
}
```

## ANEXO 2. CODIGO SISTEMA EXPERTO

```
private void ButAyudalActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Ayudal ventana1=new Ayudal(this,true);
    ventana1.setVisible(true);
}

private void ButAyuda2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Ayuda2 ventana2=new Ayuda2(this,true);
    ventana2.setVisible(true);
}

private void ButAyuda3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Ayuda3 ventana3=new Ayuda3(this,true);
    ventana3.setVisible(true);
}

private void ButAyuda4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Ayuda4 ventana4=new Ayuda4(this,true);
    ventana4.setVisible(true);
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new SistExp().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton ButAyudal;
private javax.swing.JButton ButAyuda2;
private javax.swing.JButton ButAyuda3;
private javax.swing.JButton ButAyuda4;
private javax.swing.JButton ButEvaluar;
private javax.swing.JComboBox CmbPreg1;
private javax.swing.JComboBox CmbPreg2;
private javax.swing.JComboBox CmbPreg3;
private javax.swing.JComboBox CmbPreg4;
private javax.swing.JComboBox CmbPreg5;
private javax.swing.JComboBox CmbPreg6;
private javax.swing.JLabel LabPreg1;
private javax.swing.JLabel LabPreg2;
private javax.swing.JLabel LabPreg3;
private javax.swing.JLabel LabPreg31;
private javax.swing.JLabel LabPreg4;
private javax.swing.JLabel LabPreg5;
private javax.swing.JLabel LabPreg6;
private javax.swing.JLabel LabPreg61;
// End of variables declaration
}
```



## ANEXO 2. CODIGO SISTEMA EXPERTO

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package sistemaexperto;

/**
 *
 * @author AB
 */
public class Mecanismo {

    public String Metodo(int r1,int r2, int r3, int r4, int r5, int r6){
        int res1, res2, res3, res4, res5, res6;
        String resultado=null;

        //Variables que reciben las respuestas
        res1=r1;
        res2=r2;
        res3=r3;
        res4=r4;
        res5=r5;
        res6=r6;

        //Mecanismo de Inferencia
        if (res1==2){
            resultado="Se requiere cable consola RJ-45-DB9 y computadora con Hyper Terminal";
        }
        else{
            if(res1==1 && res2==1 && res3==1 && res4==1 && res5==1 && res6==1){
                resultado="Cualquiera de los métodos NAT, CIDR o PIV6 puede ayudar";
            }
            else{
                if(res1==1 && res5==1 && res6==1 && res1==1 && res2==1 && res3==1){
                    resultado="El método que se puede implementar es: IPv6 y CIDR";
                }
                else{
                    if(res1==1 && res2==1 && res3==1 && res1==1 && res4==1){
                        resultado="El método que se puede implementar es: IPv6 y NAT";
                    }
                    else{
                        if(res1==1 && res5==1 && res6==1 &&res1==1 && res4==1){
                            resultado="El método que se puede implementar es: CIDR y NAT";
                        }
                        else{
                            if(res1==1 && res5==1 && res6==1){
                                resultado="El método que se puede implementar es: CIDR";
                            }
                            else{
                                if(res1==1 && res2==1 && res3==1){
                                    resultado="El método que se puede implementar es: IPv6";
                                }
                                else{
                                    if(res1==1 && res4==1){
                                        resultado="El método que se puede implementar es: NAT";
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        return resultado;
    }
}
```

# ANEXO 1. MANUAL DE USUARIO

## CALCULADORA DE SUPERREDES



CalculadoraCIDR.jar y

Para ejecutar la Calculadora CIDR, hay que dar doble clic al icono se mostrará la siguiente pantalla:

Calculadora CIDR

Dirección IP bloque CIDR: 0 . 0 . 0 . 0 Clase de red: A

Mascara CIDR: 128.0.0.0 Netmask: / 1

N° Máximo de Host: 2,147,483,648

Notación CIDR: Wildcard:

Rango:

Red CIDR:

Host Min:

Host Max:

Broadcast:

Figura 1. Pantalla de la calculadora CIDR

Como se puede apreciar en la Figura 1, hay datos de entrada que podemos llenar (ej. Dirección IP bloque CIDR) y otros en los que la información ya se encuentra registrada (ej. Mascara CIDR), sin embargo, también la podemos manipular. Y los datos de salida (ej. Rango o Notación CIDR) se mostrarán después de haber ingresado los datos de entrada.

Para realizar el cálculo de una superred, se tendrán que llevar a cabo los siguientes pasos:

1. Ingresar la dirección IP de la red que se quiera hacer superred.

Calculadora CIDR

Dirección IP bloque CIDR: 198 . 64 . 32 . 4 Clase de red: C

Figura 2. Dirección IP bloque CIDR y Clase de red.

Después de ingresar la dirección IP, se verifica el primer octeto y se muestra la clase de red a la que pertenece.

2. Se elige la Mascara CIDR, el numero máximo de Host (N° Máximo de Host) o se ingresa el numero de la Netmask.

Mascara CIDR: 128.0.0.0 Netmask: / 1

N° Maximo de Host: 2,147,483,648

Figura 3. Mascara CIDR, Netmask y N° Máximo de Host.

En el caso de la Mascara CIDR y N° Máximo de Host se puede elegir una opción de las que se tienen establecidas.

Mascara CIDR: 128.0.0.0 Netmask: / 1

N° Maximo de Host: 2,147,483,648

Notación CIDR: [ ] Wildcard: [ ]

Rango: [ ]

Options in dropdown: 128.0.0.0, 192.0.0.0, 224.0.0.0, 240.0.0.0, 248.0.0.0, 252.0.0.0, 254.0.0.0, 255.0.0.0

Figura 4. Opciones de Mascara CIDR.

N° Máximo de Host: 2,147,483,648

Notación CIDR: [ ] Wildcard: [ ]

Rango: [ ]

Red CIDR: [ ]

Host Min: [ ]

Options in dropdown: 2,147,483,648, 1,073,741,822, 536,870,910, 268,435,454, 134,217,726, 67,108,862, 33,554,430, 16,777,214

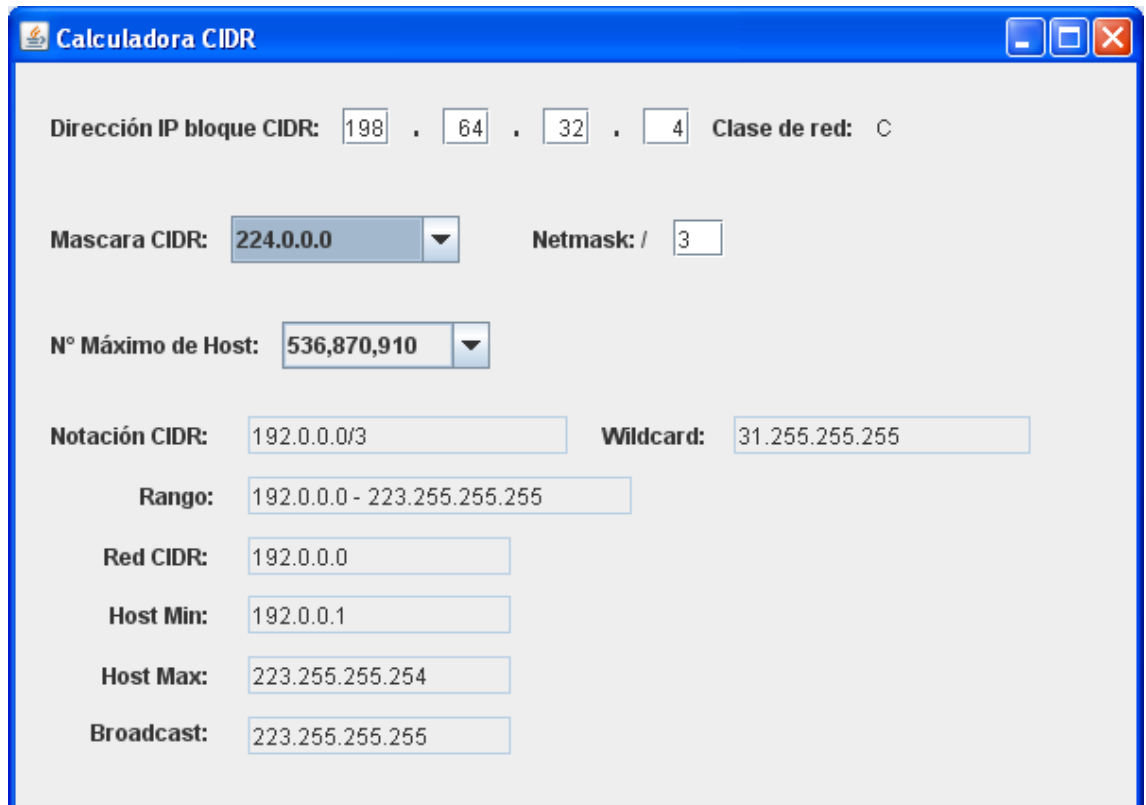
Figura 5. Opciones N° Máximo de Host.

Para la Netmask, hay que ingresar un número del 1 al 31 y posteriormente la tecla Enter, para que se pueda registrar y llevar acabo el cálculo.

Mascara CIDR: 128.0.0.0 Netmask: / 18

Figura 6. Netmask.

- Una vez ingresado alguno de los tres datos, se realizara el cálculo y se mostraran los resultados en cada uno de los datos de salida.

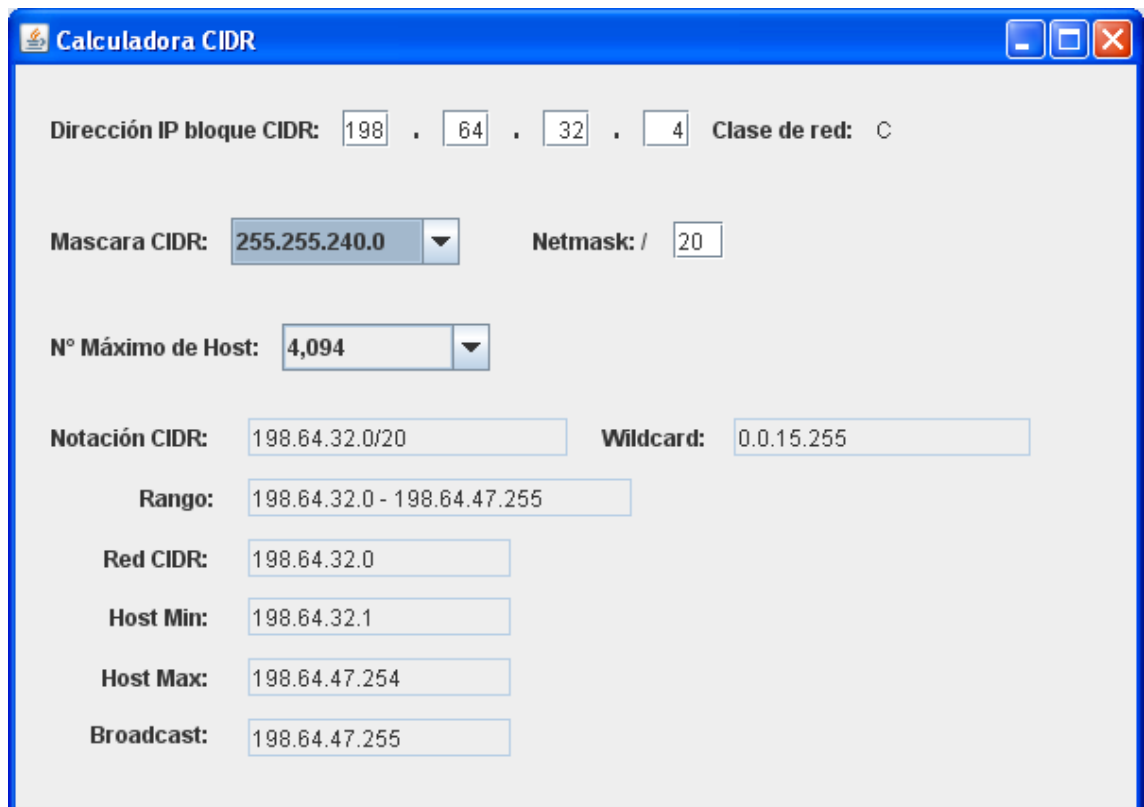


The screenshot shows a window titled "Calculadora CIDR" with the following fields and values:

- Dirección IP bloque CIDR:** 198 . 64 . 32 . 4 **Clase de red:** C
- Mascara CIDR:** 224.0.0.0 (dropdown menu)
- Netmask: /** 3
- N° Máximo de Host:** 536,870,910 (dropdown menu)
- Notación CIDR:** 192.0.0.0/3
- Wildcard:** 31.255.255.255
- Rango:** 192.0.0.0 - 223.255.255.255
- Red CIDR:** 192.0.0.0
- Host Min:** 192.0.0.1
- Host Max:** 223.255.255.254
- Broadcast:** 223.255.255.255

Figura 7. Resultados del cálculo de Superred.

Una vez mostrados los datos, se puede elegir otra Mascara CIDR o N° Máximo de Host o ingresar otra Netmask con la misma dirección.



The screenshot shows a window titled "Calculadora CIDR" with the following fields and values:

- Dirección IP bloque CIDR:** 198 . 64 . 32 . 4 **Clase de red:** C
- Mascara CIDR:** 255.255.240.0 (dropdown menu)
- Netmask:** / 20
- N° Máximo de Host:** 4,094 (dropdown menu)
- Notación CIDR:** 198.64.32.0/20
- Wildcard:** 0.0.15.255
- Rango:** 198.64.32.0 - 198.64.47.255
- Red CIDR:** 198.64.32.0
- Host Min:** 198.64.32.1
- Host Max:** 198.64.47.254
- Broadcast:** 198.64.47.255

Figura 8. Resultados de cálculo de Superred.

La misma operación se puede realizar con otra dirección IP. Y se pueden cambiar los datos constantemente.

## SISTEMA EXPERTO



Para ejecutar el Sistema Experto hay que dar doble clic al icono `SistemaExperto.jar` y se mostrará la siguiente pantalla:

1. ¿Se tiene cable de consola RJ-45-DB9 y una Computadora con HyperTerminal para configurar el rout...  
Elegir opción ▼

2. ¿Se cuenta con un router Cisco con versión de Cisco IOS 12. x T, 12.2S, 12.2SY, 12.3, 12.4, 15.0, 15.0S, 15.0SY, o 15.1? ?  
Elegir opción ▼

3. ¿Se cuenta con Sistema Operativo en los host como: Windows XP con SP1 o superior, Windows 7, Windows Vista, Windows Server 2003, Windows Server 2008, Linux con Kernel 2.2 o superior o Mac OS X 10.6 o superior?  
Elegir opción ▼

4. ¿Se cuenta con un servidor DHCP? ?  
Elegir opción ▼

5. ¿Se tiene una red clase A, B o C? ?  
Elegir opción ▼

6. ¿Se cuenta con protocolos de enrutamiento sin clases como: RIPv2, EIGRP, OSPF o BGPv4 y más de una red contigua a la utilizada? ?  
Elegir opción ▼

**Evaluar**

Figura 1. Interfaz de Sistema Experto.

El funcionamiento del Sistema Experto es el siguiente:

1. Contestar cada pregunta, eligiendo una respuesta entre SI o NO.

1. ¿Se tiene cable de consola RJ-45-DB9 y una Computadora con HyperTerminal para configurar el rout...  
Elegir opción ▼  
Elegir opción  
Si  
No

un router Cisco con versión de Cisco IOS 12. x T, 12.2S, 12.2SY, 12.3, 12.4, 15.0, 15.0S, 15.0SY, o 15.1? ?

Figura 2. Opciones de respuesta.



2. Al finalizar, presionar el botón EVALUAR, para que se muestre el resultado.

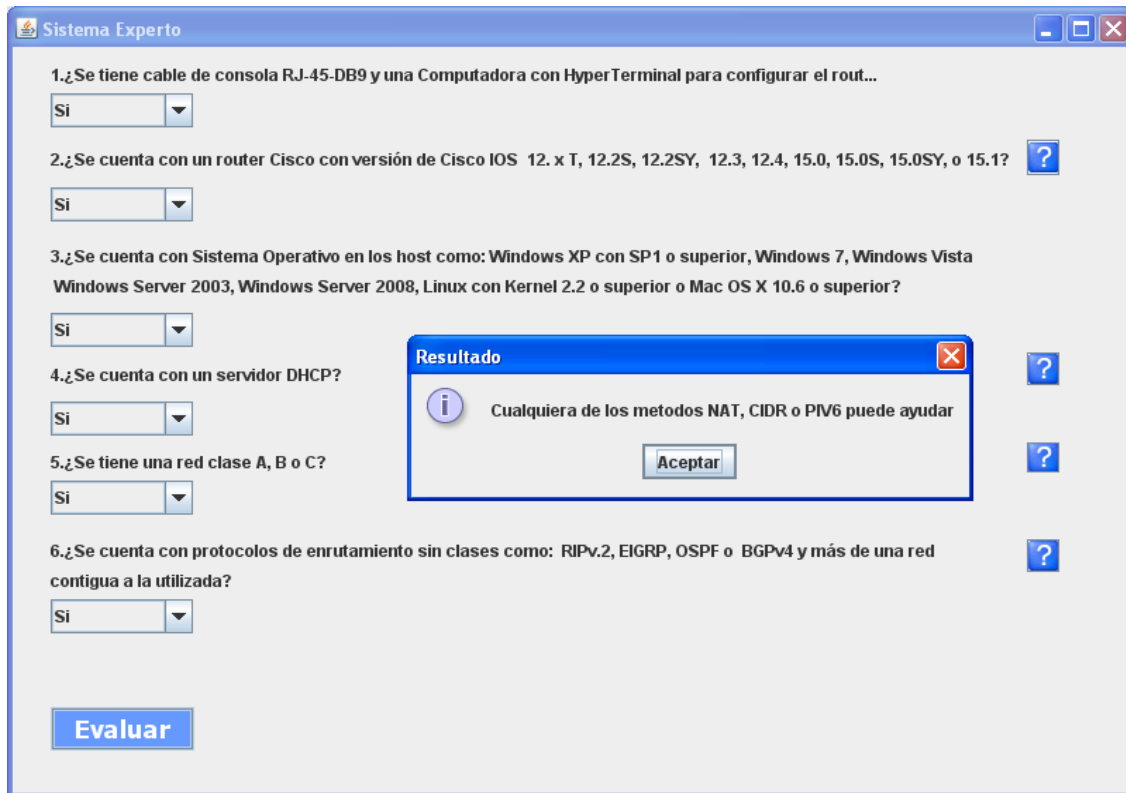


Figura 3. Ventana de Respuesta.

Para volver a contestar las preguntas o cerrar la aplicación, se presiona el botón Aceptar de la ventana emergente o se presiona la X.

A un costado de algunas preguntas, se encuentran unos botones de ayuda (Figura 4), que al presionarlo, muestra una ventana de ayuda (Figura 5), en la que se da una sugerencia para obtener la información que se pregunta.



Figura 4. Botón de ayuda.

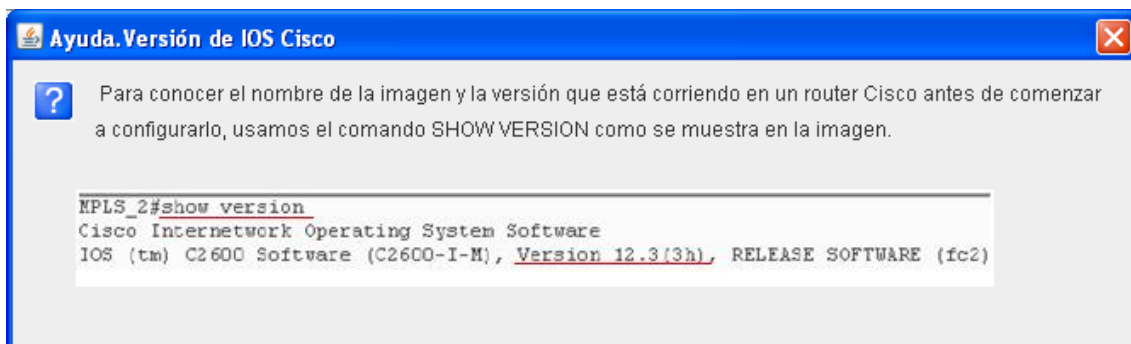
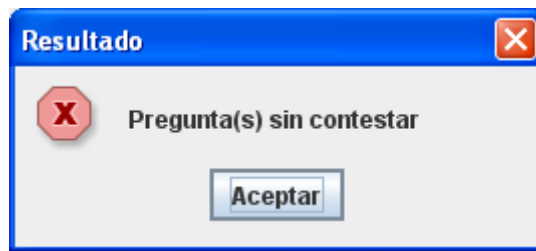


Figura 5. Ventana de Ayuda.

Para seguir contestando las preguntas, la ventana se cierra y regresamos a la venta principal.

En caso de no contestar alguna pregunta, aparecerá un mensaje de error.



**Universidad Autónoma Metropolitana**  
**Unidad Azcapotzalco**  
División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación

Proyecto Terminal

**El problema de agotamiento de las direcciones lógicas y un método para su resolución en IPv.4**

Alumno:

**Abraham Browder Serrano 205360457**

Trimestre 12-P

Fecha de entrega  
17-07-2012

Profesor Asesor:

**Rossen Petrov Popnikolov Potchinkov**  
Titular C  
Departamento de Electrónica

## **OBJETIVO GENERAL**

- Desarrollar aplicaciones que ofrezcan posibles soluciones al problema de agotamiento de las direcciones lógicas IPv4, utilizando los métodos CIDR, NAT e IPv6.

## **OBJETIVOS ESPECIFICOS**

- Diseñar e implementar un sistema experto, que proporcione una solución al problema de agotamiento de las direcciones lógicas en IPv4, utilizando los métodos CIDR, NAT e IPv6.
- Diseñar y desarrollar la Base de Conocimiento para el sistema experto.
- Diseñar y desarrollar el Mecanismo de Inferencia para el sistema experto.
- Implementar una interfaz grafica para el sistema experto.
- Desarrollar una herramienta para el cálculo de CIDR o superredes.
- Implementar una interfaz grafica para la herramienta calculadora de superredes.

## **INTRODUCCIÓN**

En el trabajo a desarrollar, se planteará la problemática que existe con el agotamiento de las direcciones lógicas IPv4. Y algunos métodos que ayudarán a solucionar el problema como: la nueva versión de IP (IPv6); el uso de direcciones privadas, con su conversión NAT a direcciones públicas y el uso de superredes, para el acoplamiento de varias redes en una sola.

Para ayudar a resolver este problema, se diseñará un sistema experto, para elegir una solución, tomando en cuenta las condiciones donde se desee emplear y el procedimiento para aplicarlo. Además profundizará en el tema de CIDR o superredes, con el diseño de una aplicación que ayude a calcularla.

## **JUSTIFICACION**

Con cientos de usuarios, en decenas de miles de localidades alrededor del mundo que dependen de la red global de Internet como parte de su ambiente diario de trabajo o de esparcimiento y diversión, puede parecer que Internet es una infraestructura de producción estable. Nuevos grupos conectan sus redes y descubren nuevas formas de utilizar la tecnología.

Es inminente el agotamiento del espacio de direcciones, que tiene una longitud de 32 bits, para la asignación de una dirección lógica y que en su momento era más que suficiente. Hoy en día el aumento de la demanda de direcciones IP ha superado las expectativas, por lo que se esta en busca de soluciones al problema.

Los métodos básicos para solucionar el problema de agotamiento de direcciones son:

- 1) IPv6. Cuenta con un nuevo tamaño en la longitud de las direcciones, que va de 32 bits a los 128 bits. Lo hace tan largo, que cada persona en el planeta podría tener una red de redes tan extensa como el Internet actual. La desventaja es que en muchos grupos no se cuenta con la infraestructura para poder implementarlo y hacer el cambio. El método a pesar de traer algunos otros beneficios como mejor calidad de servicio y altas velocidades, es algo que necesita tiempo para adaptarse y llegar a sustituir a IPv4.

- 2) NAT (Network Address Translation). Es un mecanismo para conservar direcciones IP registradas en las grandes redes y simplificar las tareas de administración de direccionamiento IP, al convertir una dirección privada en pública. Al encaminar un paquete de una red privada al exterior como Internet, a través de un dispositivo fronterizo como un router, la dirección fuente que es una dirección IP privada, se traduce a una dirección pública de internet encaminada fuera de la red privada.

Sin embargo NAT aumenta el retardo, que se introduce al efectuar la traducción de la dirección IP en el encabezado del paquete. Se observa una pérdida de la posibilidad de rastreo IP de extremo a extremo. Representa una ventaja para la seguridad pero desventaja para las aplicaciones que usen direcciones IP locales, en lugar de nombres de dominio.

- 3) Superredes o CIDR. Permite la utilización de muchas direcciones IP de clase "C" para que se vean como una sola red. De esta forma se aprovechan este tipo de redes. Hoy en día casi nadie quiere usar las redes de clase C, de las cuales existen más de 2 millones. Con el esquema de direccionamiento de superred se aprovecharían el tipo de redes clase "C".

Se tiene en cuenta que la solución próxima al problema es el uso de las direcciones IPv6, sin embargo, es un método que a pesar de que se encuentra activo en algunos lugares, no ha sido posible implementarlo en todo el mundo en sustitución de IPv4, por su alto costo.

Es de conocimiento general que las soluciones mencionadas ya se emplean actualmente. Lo que se pretende hacer en este proyecto, es diseñar un sistema experto que ayudará a decidir cuál de los métodos es el mejor para usar en cada caso. Y con el desarrollo de una aplicación para el cálculo de superredes, se pretende facilitar el uso del método y explotarlo como solución. Y así optimizar el uso de cada uno de ellos.

## ANTECEDENTES

Después de haber hecho una búsqueda exhaustiva en los PT concluidos, se puede afirmar que dentro de la UAM Azcapotzalco no hay un trabajado sobre la problemática del agotamiento de las direcciones lógicas en IPv4. Existen trabajos sobre redes, que desarrollan otros aspectos del tema. El más parecido, es un trabajo que está en desarrollo sobre "Geo-Localizador con Trace Router y cálculo de subredes" del trimestre 11-I.

En la tabla 1, se puede observar los puntos que se desarrollan en cada trabajo y la similitud o diferencias en las herramientas.

TABLA COMPARATIVA DE HERRAMIENTAS	
Geo-Localizador con Trace Router y cálculo de subredes	<b>El problema de agotamiento de las direcciones lógicas y un método para su resolución en IPv4</b>
Cálculo de subredes clase "A", "B" y "C"	Cálculo de dirección CIDR clase "C"
Cálculo de posible máscara	Cálculo de rango de direcciones de superred
Proporciona subred, número de nodos y broadcast	Calcula número de direcciones obtenidas con la superred

*Tabla 1. Comparación de cada herramienta.*

Sobre fuentes externas que se encuentran en Internet, algunos trabajos sobre el tema del agotamiento de direcciones IP, se tiene la tesis de “Análisis y diseño de una reingeniería organizativa de la red del campus de la universidad técnica de Manabí mediante la utilización de IPv.6 y su implementación en la Facultad de Ciencias Informáticas en el laboratorio de redes” [1]. Desarrolla el cambio de IPv4 a IPv6 en el laboratorio de redes en la Facultad de Ciencias Informáticas, como solución al problema de agotamiento de direcciones lógicas. No se mencionan otros métodos como el de NAT o CIDR como métodos de solución o sobre la creación de alguna herramienta, como lo que se va realizar en este trabajo.

Y en cuanto a calculadoras de superredes, se tienen algunos sitios en Internet que cuentan con esta herramienta, pero no es posible su portabilidad y la información que proporciona no es la suficiente. Tal es el caso de la calculadora “IPcalc” [2]. Y otras como “CIDR Network Calculator” [3], que es un software para teléfonos móviles y el cual tiene un costo elevado.



## DESCRIPCIÓN TÉCNICA

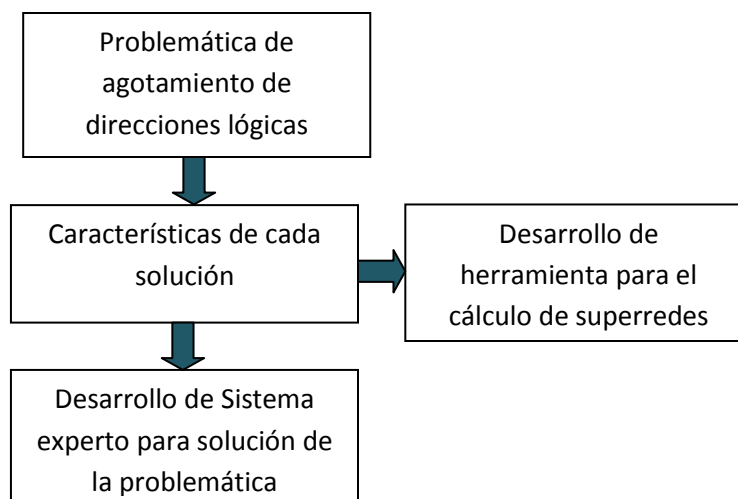


Diagrama 1. Descripción del proyecto.

### Problema de agotamiento

La experiencia ha demostrado que la división del espacio de direcciones numéricas en *clases A, B y C* ha resultado ser bastante inflexible e ineficiente en muchos casos.

En la imagen 1, se observa el rango de direcciones que se pueden asignar, dependiendo el tipo de clase y la cantidad de Hosts por clase.

Clase	Rango	Nº de Redes	Nº de Host Por Red	Máscara de Red	Broadcast ID
A	0.0.0.0 - 127.255.255.255	128	16.777.214	255.0.0.0	x.255.255.255
B	128.0.0.0 - 191.255.255.255	16.384	65.534	255.255.0.0	x.x.255.255
C	192.0.0.0 - 223.255.255.255	2.097.150	254	255.255.255.0	x.x.x.255
(D)	224.0.0.0 - 239.255.255.255	multicast			
(E)	240.0.0.0 - 255.255.255.255	reservado			

Imagen 1. Tabla de clases.

Las clases generan un problema de poca granularidad, por lo que se produce un gran desperdicio. Por ejemplo:

- Si una empresa necesita 50 direcciones lógicas, se le asigna una dirección C.
- Si una empresa necesita 300 direcciones lógicas, se le tiene que asignar una dirección clase B, con el desperdicio de más de 65,000 direcciones.

## Soluciones y características al problema

Para aprovechar la organización de las direcciones IP y conseguir un máximo aprovechamiento, se tienen los siguientes métodos como soluciones: la creación de IPv6, el uso de direcciones privadas y su conversión de direcciones públicas con NAT y la creación de superredes o CIDR.

- 1) La función de la dirección IPv6 es exactamente la misma a su predecesor IPv4, pero dentro del protocolo IPv6.

Está compuesto por 8 segmentos de 2 bytes cada uno, que suman un total de 128 bits, el equivalente a unos 3529 hosts direccionales aproximadamente. La ventaja con respecto a la dirección IPv4 es obvia en cuanto a su capacidad de direccionamiento.

Su representación suele ser hexadecimal y para la separación de cada par de octetos se emplea el símbolo “:”. Un bloque abarca desde 0000 hasta FFFF.

- 2) Con el uso de NAT (Imagen 2) temporalmente se alivia la falta de estas direcciones de red.

Este mecanismo consiste en usar una sola dirección IPv4 pública para que una red completa pueda acceder a internet. Pero esta solución nos impide la utilización de varias aplicaciones, ya que sus protocolos no son capaces de atravesar los dispositivos NAT, por ejemplo, algunas aplicaciones como: voz sobre IP (VoIP), juegos multiusuario, entre otros.

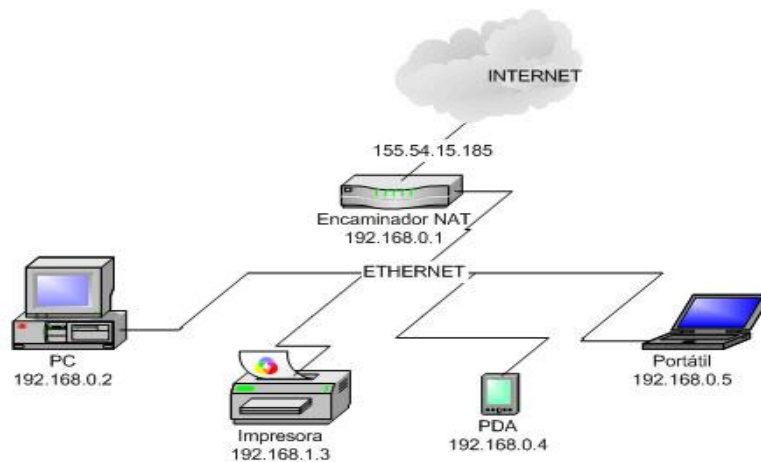


Imagen 2. Esquema del funcionamiento de NAT.

- 3) Superredes como una posible solución.

Para entender el concepto de CIDR o superred y saber como funciona, es necesario definir antes, los tipos de direccionamiento IP.

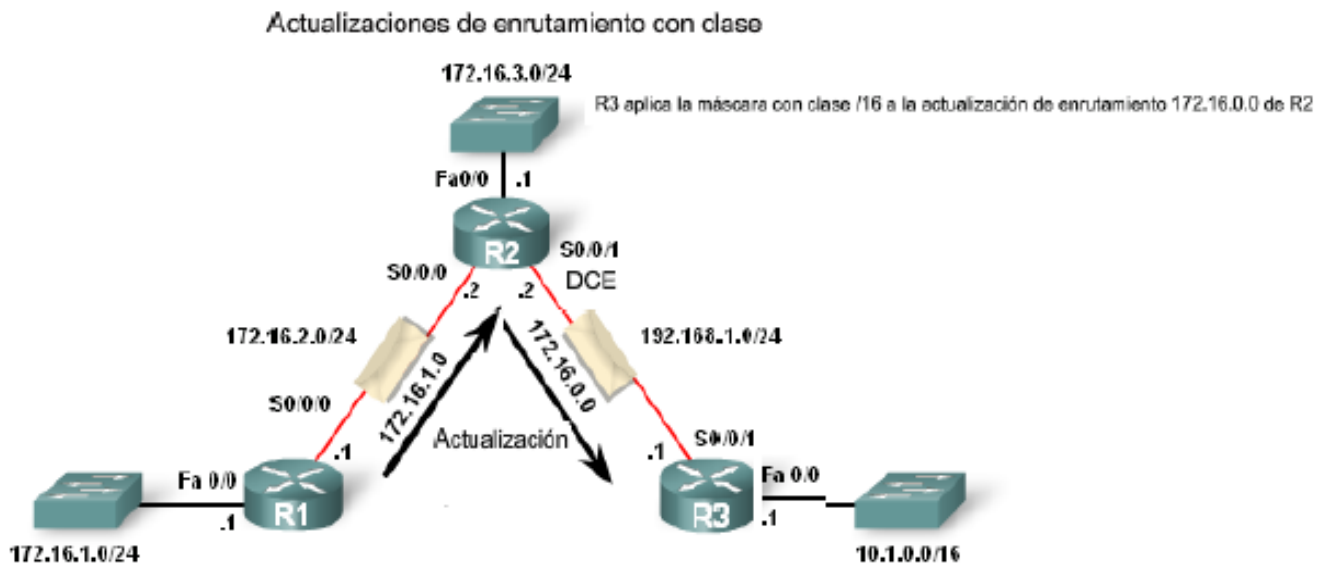
Se tiene dos formas distintas de direccionamiento: **el direccionamiento con clase (classful)** y **el direccionamiento sin clase (classless)**.

### *Direcciones con clase*

El esquema de direccionamiento IP original está conformado por tres clases principales de direcciones, A, B y C. Se diferencian según que tantos octetos se utilizan para la identificación de la red y que tantos para el ID de host. Por ejemplo, las direcciones de clase C dedican 24 bits para el ID de red y 8 para el ID de host.

En el caso de direccionamiento de subredes, la división de dos niveles red/host de la dirección IP se convierte en un sistema de tres niveles mediante la adopción de un número de bits del ID de host de clase A, B o C para usarlo como identificador de red. El identificador de red no se modifica. El ID de subred se utiliza para el enrutamiento dentro de las diferentes subredes que constituyen una red completa, proporcionando una mayor flexibilidad para los administradores. Por ejemplo, considere una dirección de clase C que normalmente utiliza los primeros 24 bits para el ID de red y los restantes 8 bits para el ID de host. El ID de host puede dividirse para un ID de subred y el ID de host.

Los protocolos de enrutamiento classful (es decir, RIPv1) no envían máscaras de subred en las actualizaciones de enrutamiento. La razón de esto es que la máscara de subred se relaciona directamente con la dirección de red.



*Imagen 3. Descripción de actualizaciones de enrutamiento con clase.*

### ***Direcciones sin clases***

Este esquema fue introducido hacia la mitad de la década de 1990. En el esquema de direcciones sin clase, la parte que identifica a la red dentro de la dirección IP puede comprender cualquier número de bits, en lugar de estar restringido a los límites impuestos por las clases fijas. Una dirección sin clase se puede representar en notación decimal de punto de forma  $w.x.y.z/n$ , donde  $n$  es el número de máscara o el número de bits que ocupa la parte de identificación de red dentro de la dirección. Es importante observar que, aunque el esquema de direcciones sin clase permite un uso más eficiente del espacio de direcciones.

### ***Características del protocolo de enrutamiento classless:***

- Las actualizaciones de enrutamiento incluyen la máscara de subred.
- Soporta VLSM
- Soporta la sumarización de ruta por CIDR

### ***CIDR***

***Classless Inter-Domain Routing*** o CIDR (enrutamiento entre dominios sin clases) se introdujo para mejorar tanto la utilización del espacio de direcciones y la escalabilidad de enrutamiento en Internet. Hacía falta, por el rápido crecimiento de Internet y el crecimiento de las tablas de enrutamiento IP existentes en los routers de Internet.

En años pasados se adoptó un esquema o notación en el que se utiliza una longitud de prefijo común arbitraria para indicar la dirección común de red de un bloque de direcciones de red contiguas que se quieren resumir en una sola dirección de red. Este esquema o notación es lo que se conoce como ***formato CIDR o de superred*** y que representa una alternativa al direccionamiento IP con clase.

En la técnica utilizada con CIDR, la máscara de red se utiliza para indicar el punto de división entre el NET ID (identificación de red) y el HOST ID (identificación de host) dentro de la dirección IP completa.

Cada uno de los routers dentro de una red de gran tamaño almacenará una copia de las máscaras de red asociadas a cada una de las redes que constituyen la red mayor, junto con la dirección de la red correspondiente. De esta forma, cuando un router recibe un paquete, primero lee la dirección IP destino de la cabecera del paquete y a continuación realiza una operación AND entre la dirección y cada una de las máscaras guardadas en su tabla. Cuando se detecta una concordancia, es decir, el net id resultante coincide con el net id asociado a la máscara de red, el router utiliza el net id y el protocolo de enrutamiento correspondiente para reenviar el paquete hacia el siguiente router dentro del camino que conduce a la red destino. Así mismo, cada Gateway debe guardar una copia de su propia máscara de red, de manera que usando dicha máscara, primero extrae la dirección IP del destinatario y después la utiliza para encaminar el paquete al host.

En la práctica CIDR requiere que cada grupo de direcciones sea una potencia de dos y utilizar una máscara de bit para identificar el tamaño del grupo.

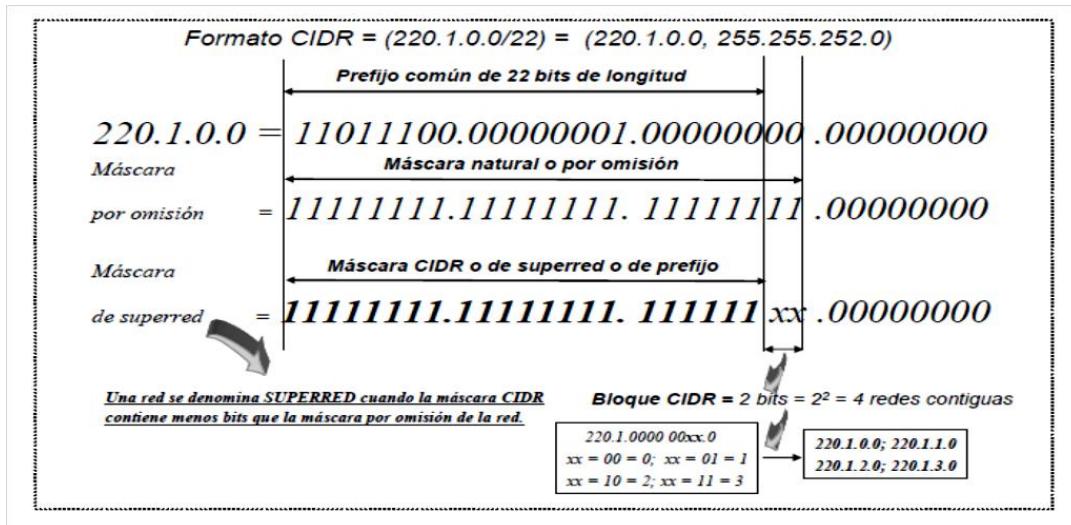


Imagen 4. Ejemplo de superred.

En el prefijo común formado por un número de bits contiguos y comunes, situados más a la izquierda, y que comparten todas las direcciones adyacentes que se desean resumir en una sola dirección. Dicho de otra forma, define el número de bits que se van a usar como prefijo común del bloque contiguo de direcciones, cuya primera dirección de red es especificada.

### Sumarización de ruta realizada por CIDR

El **resumen de rutas**, también llamado **sumarización o agregación de rutas, upernetting, superredes**, etc. es un proceso que realizan los routers por el cual toman un grupo de direcciones de redes contiguas (bloque CIDR) y las resumen en una sola dirección de red común a todas esas redes.

La principal ventaja del resumen de rutas es la optimización del enrutamiento en Internet y grandes redes corporativas, ya que los routers tienen que mantener menos entradas en sus tablas de enrutamiento y en consecuencia se gana en estabilidad, ahorro de recursos, eficiencia y tiempos de proceso. Si un router tiene conectadas 10 redes contiguas, solo publicará el resumen de ruta CIDR a sus vecinos.

Hay que tener en cuenta que solo los protocolos sin clase como: EIGRP, OSPF, RIP v.2, IS-IS y BGP soportan resumen de rutas.

Para obtener la dirección de resumen de ruta se toman las direcciones, se organizan de menor a mayor, se asegura que son direcciones contiguas, se pasa a binario y se coteja bit por bit con la operación lógica AND.

	1	1	0	0
AND	1	0	1	0
	1	0	0	0

Imagen 5. Ejemplo de operación AND.

Los bits “0” y “1” que sean comunes a todas las direcciones van a ser la dirección IP de la ruta resumida. Al mismo tiempo la suma de la cantidad de bits “0” y “1” que son comunes a todas las direcciones van a ser la máscara.

Ejemplo 1. Resumen de rutas

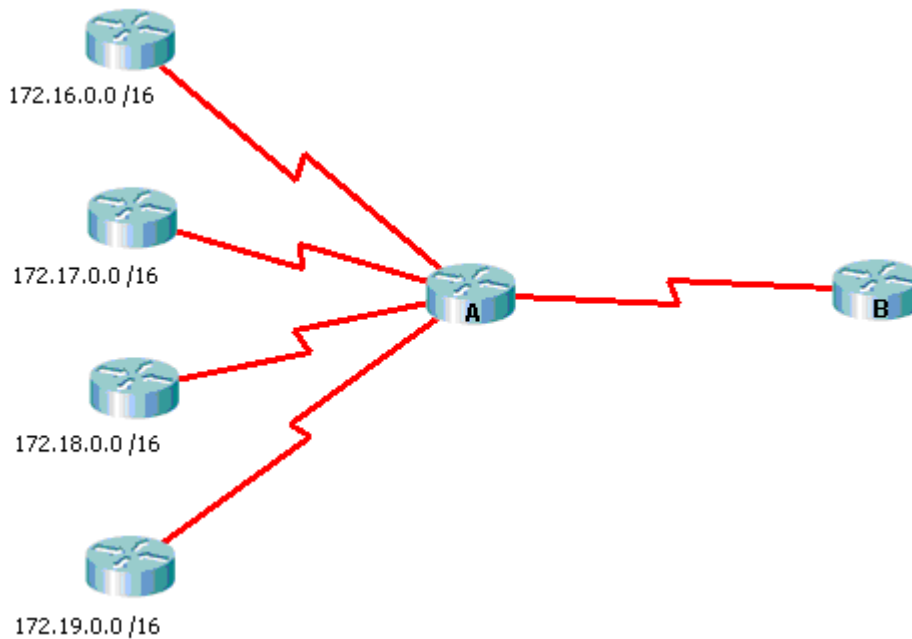


Imagen 6. Resumen de rutas.

El Router A sin usar resumen de rutas debería enviar al Router B las direcciones de las 4 redes conectadas. Nosotros debemos averiguar cuál va a ser la dirección de resumen de ruta que debe enviar el Router A al Router B si realiza sumarización.

Pasos para calcular una ruta sumariada:

- Enumere las redes en formato binario
- Cuente la cantidad de bits a la izquierda que más coincidan para determinar la máscara de la ruta sumariada
- Copie los bits coincidentes y agregue bits cero para determinar la dirección de red sumariada

```

172.16.0.0 = 10101100.00010000.00000000.00000000
172.17.0.0 = 10101100.00010001.00000000.00000000
172.18.0.0 = 10101100.00010010.00000000.00000000
172.19.0.0 = 10101100.00010011.00000000.00000000
-----
10101100.00010000.00000000.00000000
 172      16      0      0
  
```

*Imagen 7. Sumarización de direcciones.*

Para obtener la máscara sumamos los bits comunes a todas las direcciones de red.

```

172.16.0.0 = 10101100.00010000.00000000.00000000
172.17.0.0 = 10101100.00010001.00000000.00000000
172.18.0.0 = 10101100.00010010.00000000.00000000
172.19.0.0 = 10101100.00010011.00000000.00000000
-----
10101100.00010000.00000000.00000000
 14 bits comunes
  
```

*Imagen 8. Sumarización de máscaras.*

Es decir que el resumen de rutas que va a enviar el router A, que contiene las 4 redes, al Router B va a ser **172.16.0.0 /14** (/14 = 255.252.0.0).

## Ejemplo 2. Resumen de Rutas

Este ejemplo es un poco más complejo que el anterior. Se tiene esta red subneteada con VLSM y se debe averiguar cuál va a ser la dirección que va a publicar en Internet el router

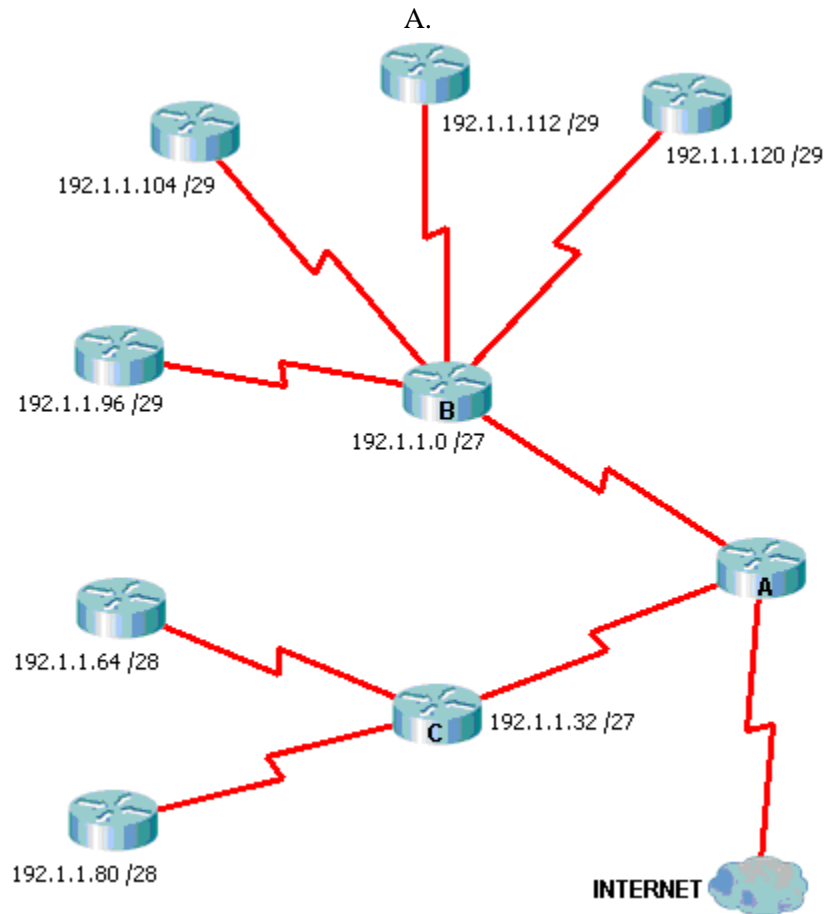


Imagen 9. Resumen de rutas.

Como se realizó en el ejercicio anterior de resumen de rutas, se organiza las direcciones de red de menor a mayor, se convierten a binario, se hace un AND y se obtiene la dirección de la ruta resumida.



```

192.1.1.0 = 11000000.00000001.00000001.00000000
192.1.1.32 = 11000000.00000001.00000001.00100000
192.1.1.64 = 11000000.00000001.00000001.01000000
192.1.1.80 = 11000000.00000001.00000001.01010000
192.1.1.96 = 11000000.00000001.00000001.01100000
192.1.1.104 = 11000000.00000001.00000001.01101000
192.1.1.112 = 11000000.00000001.00000001.01110000
192.1.1.120 = 11000000.00000001.00000001.01111000

```

---

```

11000000.00000001.00000001.00000000

```

192
1
1
0

*Imagen 10. Ruta resumida.*

Resta obtener la máscara sumando los bits “0” y “1” que son comunes a todas las direcciones, en este caso son 25 bits.

```

192.1.1.0 = 11000000.00000001.00000001.00000000
192.1.1.32 = 11000000.00000001.00000001.00100000
192.1.1.64 = 11000000.00000001.00000001.01000000
192.1.1.80 = 11000000.00000001.00000001.01010000
192.1.1.96 = 11000000.00000001.00000001.01100000
192.1.1.104 = 11000000.00000001.00000001.01101000
192.1.1.112 = 11000000.00000001.00000001.01110000
192.1.1.120 = 11000000.00000001.00000001.01111000

```

---

```

11000000.00000001.00000001.00000000

```

**25 bits comunes**

*Imagen 11. Mascara resumida.*

El resumen de rutas que va a enviar el router A a Internet que contiene la totalidad de las red de los routers B y C va a ser **192.1.1.0 /25** (/25 = 255.255.255.128).

## Desarrollo de un sistema experto para solución de las problemática de agotamiento de direcciones lógicas.

### Base de conocimiento

#### Clasificación de clases

Una dirección IP es una dirección usada con el fin de identificar una red IP. La dirección se compone de 32 bits binarios, que pueden ser dividirse en una parte de red y con la ayuda de una máscara de subred. Los 32 bits binarios se dividen en cuatro octetos (1 octeto = 8 bits). Cada octeto se convierte a decimal y son separados por un punto. Por esta razón, una dirección IP se dice que se expresan en formato decimal con puntos (por ejemplo, 172.16.81.100). El valor de cada octeto varía de 0 a 255 decimal o 00000000 a 11111111 binario.

Estos octetos se descomponen para proporcionar un servicio de direcciones que puede adaptarse a redes grandes y pequeñas. Hay cinco diferentes clases de redes, de la A a la E. Las más usadas son las clases A, B, C, ya que las clases D y E están reservadas.

Dada una dirección IP, su clase puede determinarse a partir de los tres bits de orden superior. La Figura 1 muestra la importancia de los tres bits de orden superior y el rango de direcciones que corresponden a cada clase. Para fines informativos, Clase D y direcciones de clase E se muestran también.

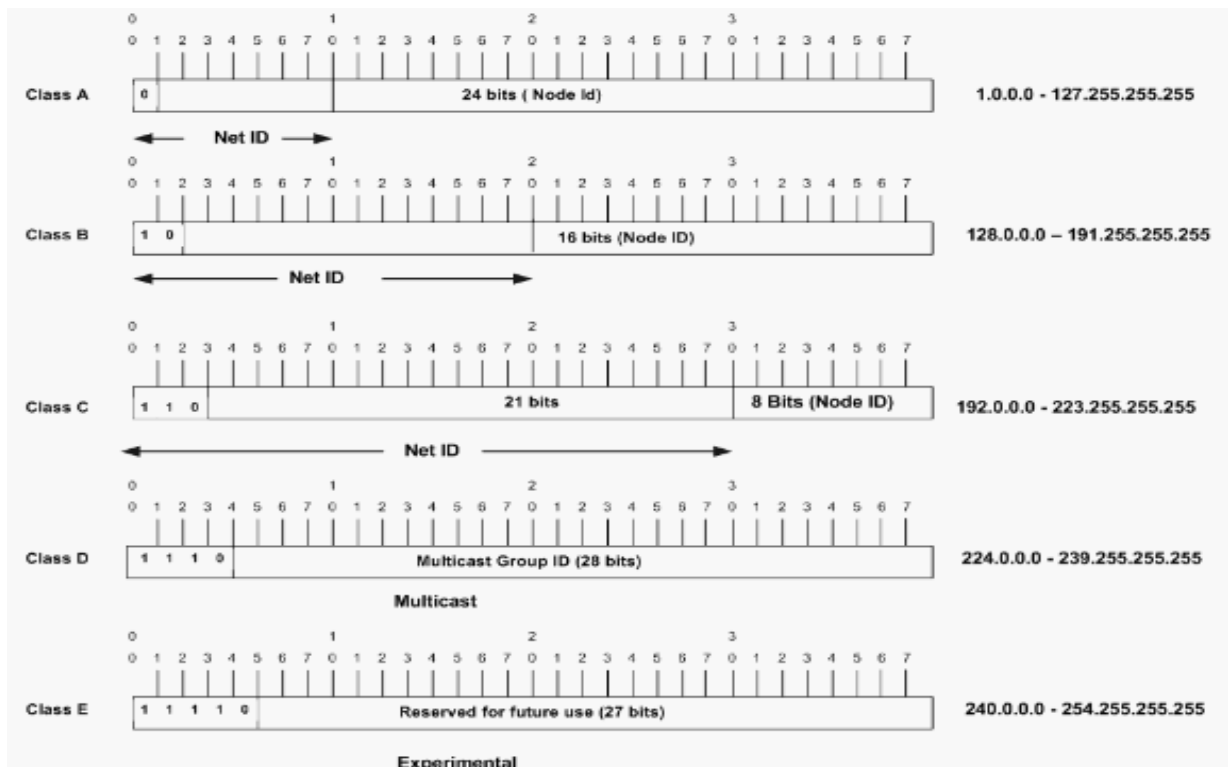


Imagen 12. Clases de red

En una dirección Clase A, el primer octeto es la porción de red. La dirección de red tiene un rango de 1.0.0.0 - 127.255.255.255. Los octetos 2, 3 y 4, (los siguientes 24 bits) son para dividir en subredes y hosts. Las direcciones de clase A se utilizan para las redes que tienen más de 65.536 hosts.

En una dirección Clase B, los dos primeros octetos son la porción de red. La dirección de red tiene un rango de 128.0.0.0 - 191.255.255.255. Los octetos 3 y 4 (16 bits) son para las subredes locales y hosts. Las direcciones de clase B se utilizan para las redes que tienen entre 256 y 65.534 hosts.

En una dirección de clase C, los primeros tres octetos son la porción de red. En el ejemplo de la Clase C tiene una dirección de red de 192.0.0.0 - 233.255.255.255. El octeto 4 (8 bits) es para las subredes locales y hosts, perfecto para las redes con menos de 254 hosts.

### ***Las máscaras de red***

Una máscara de red le ayuda a saber qué parte de la dirección identifica la red y que parte de la dirección identifica el nodo. Las redes de clase A, B y C tienen las máscaras por defecto, también conocidos como mascarar por default, como se muestra aquí:

Clase A: 255.0.0.0  
Clase B: 255.255.0.0  
Clase C: 255.255.255.0

Una dirección IP en una red de clase que no ha sido subdividida tendría un par dirección / máscara similar a: 8.20.15.1 255.0.0.0. Para ver cómo la máscara le ayuda a identificar las partes de la red y el nodo de la dirección, se convierte la dirección y la máscara a números binarios.

8.20.15.1 = 00001000.00010100.00001111.00000001  
255.0.0.0 = 11111111.00000000.00000000.00000000

Una vez que tenga la dirección y la máscara representada en binario, identificar la red y el ID de host es más fácil. Todos los bits de la dirección que tienen los bits correspondientes a los 1 de la máscara, representan el ID de red. Cualquier bit de dirección que tienen los bits correspondientes a 0 en la máscara, representan el identificador de nodo.

8.20.15.1 = 00001000.00010100.00001111.00000001  
255.0.0.0 = 11111111.00000000.00000000.00000000

-----  
Net ID | ID de host

netid = 00001000 = 8  
hostid = 00010100.00001111.00000001 = 20.15.1

## **CIDR**

En CIDR, una red IP está representada por un prefijo, que es una dirección IP y alguna indicación de la longitud de la máscara. Longitud: el número de bits de la máscara más a la izquierda contiguos que se establecen en una. Así que la red 172.16.0.0 255.255.0.0 se puede representar como 172.16.0.0/16. CIDR también se muestra una arquitectura de Internet más jerárquica, donde cada dominio tiene sus direcciones IP desde un nivel superior. Este permite el resumen de los dominios que hay que hacer en el nivel superior. Por ejemplo, si un ISP posee la red 172.16.0.0/16, el ISP puede ofrecer 172.16.1.0/24, 172.16.2.0/24, y así sucesivamente a los clientes. Sin embargo, al anunciar a otros proveedores, el ISP sólo necesita hacer publicidad de 172.16.0.0/16.

El **resumen de rutas**, también llamado **sumarización o agregación de rutas, supernetting, superredes**, etc. es un proceso que realizan los routers por el cual toman un grupo de direcciones de redes contiguas (bloque CIDR) y las resumen en una sola dirección de red común a todas esas redes.

La principal ventaja del resumen de rutas es la optimización del enrutamiento en Internet y grandes redes corporativas, ya que los routers tienen que mantener menos entradas en sus tablas de enrutamiento y en consecuencia se gana en estabilidad, ahorro de recursos, eficiencia y tiempos de proceso. Si un router tiene conectadas 10 redes contiguas, solo publicará el resumen de ruta CIDR a sus vecinos.

Los routers aprenden los unos de los otros el camino para llegar a cada destino, la forma de aprender estos caminos es el protocolo de routing. Existen dos grandes tipos de protocolos, los IGP (Interior Gateway Protocol) que se utilizan para redes privadas, normalmente son sistemas autónomos y los EGP (Exterior Gateway Protocols), los cuales se utilizan para internet. Según la regla que utilicen para realizar la ruta se suelen dividir en Vector de distancia o bien estado del enlace, existe una tercera vía llamada Vector de distancia mejorada utilizada por Cisco.

Tabla de protocolos:

Protocolo de Routing	Classfull Classless	Convergencia	Distancia Administrativa	IP v6	Multiprotocolo	Entradas	Pequeña Mediana Grande	IGP EGP	Unicast Multicast Broadcast
Rip v1	Classfull	90s/Router	120	No	Si	1000	Pequeña	IGP	Broadcast
Rip v2	Classless	90s/Router	120	No	Si	1000	Pequeña	IGP	Multicast
IGRP	Classfull	270s/Router	100	No	Si	3000	Mediana	IGP	Broadcast
EIGRP	Classless	1a vez 2s, después 0	90	Si	Si	44000	Mediana	IGP	Multicast Unicast
OSPF	Classless	6s	110	Si	No	36000	Mediana	IGP	Multicast Unicast
ISIS	Classless	4s	115	Si	Si	70000	Mediana	IGP	Multicast Unicast
E - BEP	Classless	Entre 40s y 2 días	20	Si	Si	Sin límite	Grande	EGP	Unicast
I- BEP	Classless	Entre 40s y 2 días	200	Si	Si	Sin límite	Grande	IGP	Unicast

*Tabla 2. Protocolos de enrutamiento.*

Tengan en cuenta que solo **los protocolos sin clase EIGRP, OSPF, RIP v.2, IS-IS y BGP** soportan resumen de rutas.

### **RIPv2**

RIPv2 fue descrita por primera vez en el RFC 1388 y RFC 1723 (1994), el actual RFC es 2453, escrito en noviembre de 1998. A pesar de los entornos actuales de uso de avanzados protocolos de enrutamiento como OSPF y EIGRP, todavía hay redes que utilizan RIP. La necesidad de utilizar VLSM y otros requisitos que se definen en RIPv2.

RIPv2 mejora sobre RIPv1 en la capacidad de usar VLSM, con soporte para la autenticación de ruta, y con la multidifusión de cambios de ruta. RIPv2 soporta CIDR.

## ***EIGRP***

Cisco Systems lanzó EIGRP en la década de 1990 como una evolución de IGRP hacia un protocolo de enrutamiento más escalable para grandes interconexiones de redes. EIGRP es un protocolo sin clase que permite el uso de VLSM y que soporta CIDR para la asignación de direcciones IP escalable.

## ***OSPF***

El Open Shortest Path First (OSPF), definido en el RFC 2328 , es un Interior Gateway Protocol utiliza para distribuir la información de enrutamiento dentro de un sistema autónomo único. En este trabajo se examina la forma en OSPF funciona y cómo se puede utilizar para diseñar y construir redes de grandes y complicados.

El protocolo OSPF se basa en la tecnología de estado de enlace, que es una desviación de los algoritmos de Bellman-Ford basados en vectores utilizados en protocolos de Internet tradicionales de enrutamiento como RIP. OSPF ha introducido nuevos conceptos tales como la autenticación de las actualizaciones de enrutamiento, las máscaras de subred de longitud variable (VLSM), el resumen de ruta, y así sucesivamente.

## ***IS-IS***

El Sistema Intermedio a Sistema Intermedio (IS-IS) es un protocolo de enrutamiento de Interior Gateway Protocol (IGP), estandarizado por el Internet Engineering Task Force (IETF) y de uso común en las grandes redes de proveedores de servicios. IS-IS también puede ser desplegado en las redes de empresas muy grandes. IS-IS es un protocolo de enrutamiento de estado de enlace, proporcionando una convergencia rápida y excelente escalabilidad. Al igual que todos los protocolos de estado de enlace, IS-IS es muy eficiente en su uso de ancho de banda de red.

## ***BORDER GATEWAY PROTOCOL (BGP).***

La función principal de un sistema BGP es el intercambio de información de la red de accesibilidad con otros sistemas BGP, incluyendo información sobre la lista de rutas de sistema autónomo. Esta información puede usarse para construir una gráfica de la conectividad del sistema autónomo de la que los bucles de enrutamiento pueden ser podados y con el que a nivel de sistema autónomos las decisiones de política puede ser aplicado.

Versión BGP 4 es compatible con enrutamiento entre clases (CIDR), que le permite reducir el tamaño de las tablas de enrutamiento mediante la creación de rutas totales, resultando en *superredes* . CIDR elimina el concepto de clases de red dentro de BGP y es compatible con la publicidad de los prefijos IP. Rutas CIDR pueden ser transportadas por Open Shortest Path First (OSPF) IGRP mejorado (EIGRP), y el Sistema Intermedio a Sistema Intermedio (ISIS)-IP y Routing Information Protocol (RIP).

## *NAT*

Network Address Translation (NAT) permite acceder a Internet traduciendo las direcciones privadas en direcciones IP registradas. Incrementa la seguridad y la privacidad en direccionamiento interno a uno externo.

NAT tiene varias formas de trabajar según los requisitos y la flexibilidad de que se disponga, cualquiera de ellas es sumamente importante a la hora de controlar al tráfico hacia el exterior:

**Estáticamente:** NAT permite la asignación de una a una entre las direcciones locales y las exteriores o globales.

**Dinámicamente:** NAT permite signar a una red IP interna a varias externas incluidas en un grupo o pool de direcciones.

**PAT (Port Address Translation):** Es una forma de NAT dinámica, comúnmente llamada NAT sobrecargado, que asigna varias direcciones IP internas a una sola externa. PAT utiliza números de puertos de origen únicos en la dirección global interna para distinguir entre las diferentes traducciones.

## **Terminología**

- **Dirección local Interna:** la dirección IP asignada a un host de la red interna. Esta es la dirección para configurarse como IP del nodo o la que se recibe a través del protocolo dinámico de asignación de direcciones DHCP. La dirección no una dirección IP asignada por el Network Information Center (NIC) o proveedor de servicios.
- **Dirección global interna:** Es una dirección IP asignada por el proveedor servicio que representa una o más direcciones IP locales con el mundo exterior.
- **Dirección local externa :** la dirección IP de un host externo tal como lo ve la red interna.
- **dirección global externa :** la dirección IP asignada a un host en la red exterior por el propietario. La dirección se asigna a una dirección globalmente enrutable.

## *DHCP*

**DHCP** (acrónimo de **D**ynamic **H**ost **C**onfiguration **P**rotocol), que se traduce Protocolo de configuración dinámica de servidores) es un protocolo que permite a dispositivos individuales en una red de direcciones IP obtener su propia información de configuración de red (dirección IP; máscara de sub-red, puerta de enlace, etc.) a partir de un servidor DHCP. Su propósito principal es hacer más fáciles de administrar las redes grandes. **DHCP** existe desde 1993 como protocolo estándar, y se describe a detalle en el RFC 2131.

Sin la ayuda de un servidor **DHCP**, tendrían que configurarse de forma manual cada dirección IP de cada anfitrión que pertenezca a una Red de Área Local. Si un anfitrión se traslada hacia otra ubicación donde existe otra Red de Área Local, se tendrá que configurar otra dirección IP diferente para poder unirse a esta nueva Red de Área Local. Un servidor **DHCP** entonces supervisa, y distribuye, las direcciones IP de una Red de Área Local asignando una dirección IP a cada anfitrión que se una a la Red de Área Local.

Existen tres métodos de asignación en el protocolo **DHCP**:

- **Asignación manual:** La asignación utiliza una tabla con direcciones **MAC** (acrónimo de **Media Access Control Address**, que se traduce como dirección de Control de Acceso al Medio). Sólo los anfitriones con una dirección **MAC** definida en dicha tabla recibirá el IP asignada en la misma tabla.
- **Asignación automática:** Una dirección de IP disponible dentro de un rango determinado se asigna permanentemente al host que la requiera.
- **Asignación dinámica:** Se determina arbitrariamente un rango de direcciones IP, y cada host conectado a la red está configurada para solicitar su dirección IP al servidor cuando se inicia el dispositivo de red, **utilizando un intervalo de tiempo controlable** (parámetros **default-lease-time**, y **max-lease-time**), de modo que la asignación de direcciones IP es de manera temporal, y éstas se reutilizan de forma dinámica.

### ***IP v.6***

Es una realidad que la transición a IPv6 es inminente, las direcciones IPv4 se están acabando y a partir de ahí nuestra solución es IPv6.

En pocos meses o años (todo depende del escenario de cada empresa) nos tocará migrar a IPv6.

Por lo que existen algunos mecanismos de transición o mejor llamados técnicas de interoperación de IPv4 a IPv6, para la migración o uso de ambas versiones. Estos mecanismos son:



## Dual Stack

El mecanismo de transición dual stack es común, útil y requiere al menos un host con dual stack.

Como el nombre lo dice, tener dos stacks requiere el instalar tanto el stack IPv4 como el IPv6 en un host. Esto significa que el host puede tomar decisiones acerca de cuando las conexiones deben hacerse usando IPv4 o IPv6; generalmente esto se hace basandose en la disponibilidad de la conectividad IPv6 y los registros DNS. Los stacks IPv4 e IPv6 pueden y usualmente serán completamente independientes, pues las interfaces pueden enumerarse de forma separada, habilitar y deshabilitarlas separadamente y ser tratadas como máquinas separadas.

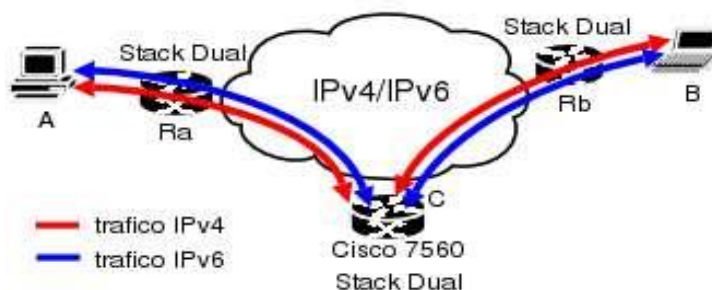


Imagen 13. Dual stack.

## Túneles Configurados

El principio detrás de la "tunelación" es el encapsular paquetes IPv6 en paquetes IPv4, es decir, empaquetar paquetes dentro de otros paquetes.

Este mecanismo es muy útil para conectar dos islas informáticas bien conocidas y muy poco probable que cambien, como pueden ser una sucursal y la oficina principal.

La idea central es entender que al igual que los encabezados ethernet rodean los paquetes IP, se puede decir que insertar otro paquete donde iría un paquete TCP y confiar en el sistema de enrutamiento lleve el paquete al destino. Si el origen como el destino saben como tratar estos paquetes, todo puede ser decodificado.

A pesar de que las funcionalidades en los routers y otros dispositivos están adecuadas para trabajar con túneles, los túneles de la forma IPv4 sobre IPv6 no han sido desarrollados en su totalidad.

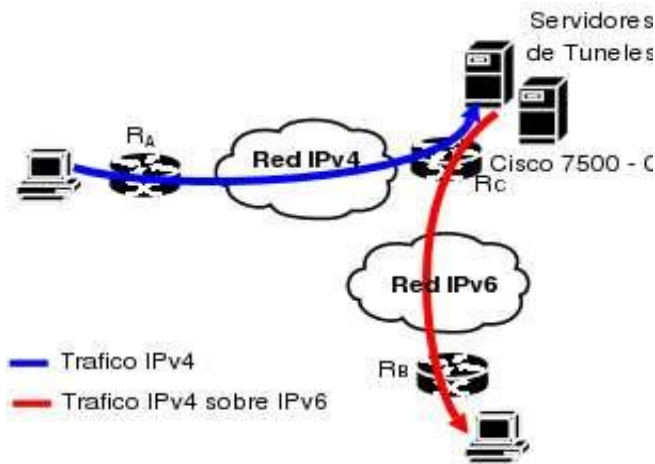


Imagen 14. Túneles Configurados.

### Cable de consola

La administración y configuración de los Router se hace regularmente por medio de un cable de consola, el cual consiste en un cable CrossOver que tiene como peculiaridad que los colores quedan invertidos.

Ahora te comparto un esquema de como hacer un cable de consola que en un extremo (el conectado al router) tiene un conector RJ-45 y en el otro extremo (el conectado a la Pc) un conector R232.

El cable que presento esta basado en norma EIA/TIA 568B.

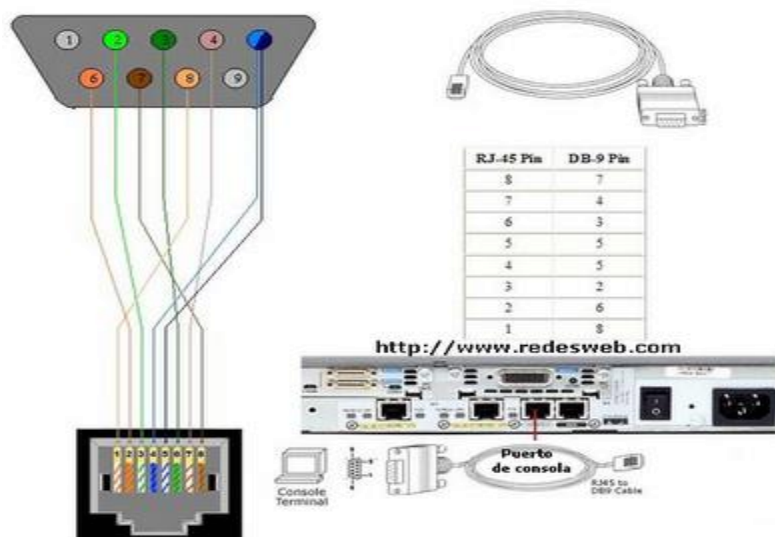


Imagen 15. Cable consola

### **Cable UTP**

El **cable de par trenzado** es un medio de conexión usado en telecomunicaciones en el que dos conductores eléctricos aislados son entrelazados para anular las interferencias de fuentes externas y diafonía de los cables adyacentes.

El entrelazado de los cables disminuye la interferencia debido a que el área de bucle entre los cables, la cual determina el acoplamiento eléctrico en la señal, se ve aumentada. En la operación de balanceado de pares, los dos cables suelen llevar señales paralelas y adyacentes (modo diferencial), las cuales son combinadas mediante sustracción en el destino. La tasa de trenzado, usualmente definida en vueltas por kilómetro, forma parte de las especificaciones de un tipo concreto de cable. Cuanto mayor es el número de vueltas, menor es la atenuación de la diafonía. Donde los pares no están trenzados, como en la mayoría de las conexiones telefónicas residenciales, un miembro del par puede estar más cercano a la fuente que el otro y, por tanto, expuesto a niveles ligeramente distintos de interferencias electromagnéticas.

La especificación *568A Commercial Building Wiring Standard* de la asociación Industrias Electrónicas e Industrias de las Telecomunicaciones (EIA/TIA) especifica el tipo de cable UTP que se utilizará en cada situación y construcción. Dependiendo de la velocidad de transmisión, ha sido dividida en diferentes categorías de acuerdo a esta tabla:

<b>Categoría</b>	<b>Ancho de banda (MHz)</b>	<b>Aplicaciones</b>	<b>Notas</b>
<b>Categoría 1</b>	0,4 MHz	Líneas telefónicas y módem de banda ancha.	No descrito en las recomendaciones del EIA/TIA. No es adecuado para sistemas modernos.
<b>Categoría 2</b>	4 MHz	Cable para conexión de antiguos terminales como el IBM 3270.	No descrito en las recomendaciones del EIA/TIA. No es adecuado para sistemas modernos.
<b>Categoría 3</b>	16 MHz	10BASE-T and 100BASE-T4 Ethernet	Descrito en la norma EIA/TIA-568. No es adecuado para transmisión de datos mayor a 16 Mbit/s.
<b>Categoría 4</b>	20 MHz	16 Mbit/s Token Ring	
<b>Categoría 5</b>	100 MHz	100BASE-TX y 1000BASE-T Ethernet	
<b>Categoría 5e</b>	100 MHz	100BASE-TX y 1000BASE-T Ethernet	Mejora del cable de Categoría 5. En la práctica es como la categoría anterior pero con mejores normas de prueba. Es adecuado para Gigabit Ethernet
<b>Categoría 6</b>	250 MHz	1000BASE-T Ethernet	Cable más comúnmente instalado en Finlandia según la norma SFS-EN 50173-1.
<b>Categoría 6e</b>	250 MHz (500MHz según otras)	10GBASE-T Ethernet (en desarrollo)	No es estandarizado. Lleva el sello del fabricante.

	fuentes)		
<b>Categoría 7</b>	600 MHz	En desarrollo. Aún sin aplicaciones.	Cable U/FTP (sin blindaje) de 4 pares.
<b>Categoría 7a</b>	1200 MHz	Para servicios de telefonía, Televisión por cable y Ethernet 1000BASE-T en el mismo cable.	Cable S/FTP (pares blindados, cable blindado trenzado) de 4 pares. Norma en desarrollo.
<b>Categoría 8</b>	1200 MHz	Norma en desarrollo. Aún sin aplicaciones.	Cable S/FTP (pares blindados, cable blindado trenzado) de 4 pares.

*Tabla 3. Categorías de cable UTP.*

Ventajas del cable UTP:

- Bajo costo en su contratación.
- Alto número de estaciones de trabajo por segmento.
- Facilidad para el rendimiento y la solución de problemas.
- Puede estar previamente cableado en un lugar o en cualquier parte.

Desventajas del cable UTP:

- Altas tasas de error a altas velocidades.
- Ancho de banda limitado.
- Baja inmunidad al ruido.
- Baja inmunidad al efecto crosstalk (diafonía).
- Alto costo de los equipos.
- Distancia limitada (100 metros por segmento).

### **Conector RJ-45**

**RJ-45** (*registered jack 45*) es una interfaz física comúnmente usada para conectar redes de cableado estructurado, (categorías 4, 5, 5e, 6 y 6a). Es parte del Código Federal de Regulaciones de Estados Unidos. Posee ocho pines o conexiones eléctricas, que normalmente se usan como extremos de cables de par trenzado.

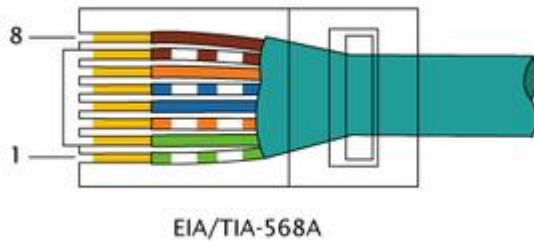
Es utilizada comúnmente con estándares como TIA/EIA-568-B, que define la disposición de los pines o *wiring pinout*.

Una aplicación común es su uso en cables de red Ethernet, donde suelen usarse 8 pines (4 pares). Otras aplicaciones incluyen terminaciones de teléfonos (4 pines o 2 pares) por ejemplo en Francia y Alemania, otros servicios de red como RDSI y T1 e incluso RS-232.

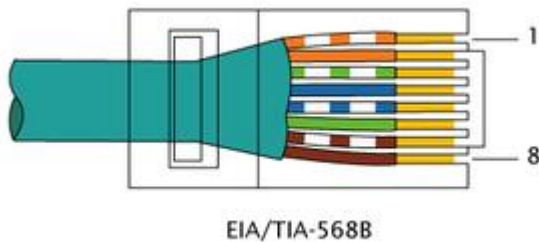
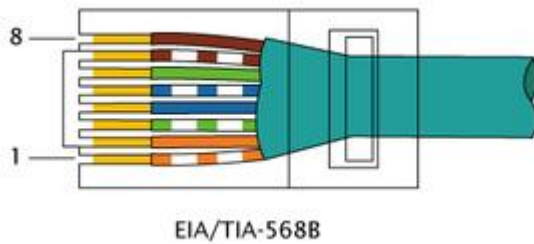
El cable directo de red sirve para conectar dispositivos desiguales, como un computador con un hub o switch. En este caso ambos extremos del cable deben tener la misma distribución. No existe diferencia alguna en la conectividad entre la distribución 568B y la distribución 568A siempre y cuando en ambos extremos se use la misma, en caso contrario hablamos de un cable cruzado.

El esquema más utilizado en la práctica es tener en ambos extremos la distribución 568B.

Cable directo 568<sup>a</sup>.



Cable directo 568B.



*Imagen 16. Esquemas de cable directo.*

### **Cable cruzado**

Un cable cruzado es un cable que interconecta todas las señales de salida en un conector con las señales de entrada en el otro conector, y viceversa; permitiendo a dos dispositivos electrónicos conectarse entre sí con una comunicación full duplex. El término se refiere - comúnmente - al cable cruzado de Ethernet, pero otros cables pueden seguir el mismo principio. También permite transmisión confiable vía una conexión ethernet.

Para crear un cable cruzado que funcione en 10/100baseT, un extremo del cable debe tener la distribución 568A y el otro 568B. Para crear un cable cruzado que funcione en 10/100/1000baseT, un extremo del cable debe tener la distribución Gigabit Ethernet (variante A), igual que la 568B, y el otro Gigabit Ethernet (variante B1). Esto se realiza para que el TX (transmisión) de un equipo esté conectado con el RX (recepción) del otro y a la inversa; así el que "habla" (transmisión) es "escuchado" (recepción).

Cable cruzado 568A/568B.

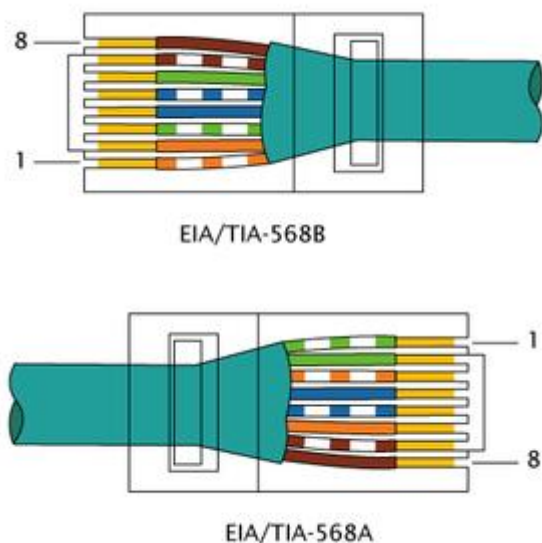


Imagen 17. Esquema de cable cruzado.

## CISCO IOS

El IOS (Internetwork Operating System) de CISCO es el sistema operativo usado en los dispositivos de red de CISCO: routers, switches, para la implementación de redes que forman la gran Internet (usando dispositivos CISCO). Una forma de acceso al CISCO IOS es usando CLI (Command Line Interface) el cual está basado en un conjunto de comandos. Estos comandos están ubicados en los diversos modos de operación definidos en cada equipo de red: usuario, privilegiado, configuración global. La selección de una adecuada versión del CISCO IOS está relacionada con las características técnicas que pueden ofrecer los equipos de red, por ejemplo el soporte de IPv6, MPLS, Seguridad, entre otros. Además, como los routers tienen diversos tipos de interfaces, siendo las más comunes Ethernet y Serie, el IOS debe contener un grupo de drivers para soportar esta variedad de interfaces. Las imágenes de CISCO IOS poseen nombres específicos y estandarizados por CISCO que reflejan sus principales características como la que plataforma soporta, las funciones realiza y la versión de IOS utiliza para facilitar su actualización. Es importante conocer el nombre de la imagen y la versión que está corriendo en un router antes de comenzar a configurarlo. Para ello usamos el comando *show versión* que se muestra a continuación.

```
MPLS_2#show version
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.3(3h), RELEASE SOFTWARE (fc2)
```

Imagen 18. Pantalla de comando show versión en router Cisco.

Típicamente el nombre del archivo que contiene a una imagen CISCO IOS consta de una parte que indica las características de la IOS y otra que corresponde a la versión de la misma. En la figura 2.1, el nombre de la imagen IOS es **C2600-I-M**, la versión que se usa es la **12.3 (3h)** y el estado de la versión es **fc2**.

### *Características del Cisco IOS*

Las características que posee una imagen IOS se indican en tres grupos de letras estandarizadas por CISCO, el primer grupo hace referencia a la plataforma, el segundo grupo a las funcionalidades de la imagen y el tercero al formato de la imagen.

Plataforma: Corresponde al primer grupo de caracteres e identifica la plataforma para la cual la imagen fue creada. La tabla 2 muestra los identificadores de plataformas más comunes.

Identificador	Plataformas
As	Access Server 5200 series
Ca	Cisco Advantage
Cpa	Cisco Pro
Cs	Communication Server
c800, c1000, c1005, c1600	Cisco 800, 1003, 1004, 1005, 1600
c2500	Cisco 2500-2525 series routers
c2600	Cisco 2610-2613 series routers
c2800	Catalyst 2800
c2900	Cisco 2910, 2950
c29atm	Cisco 2900 ATM
c3620, c3640, c3800	Cisco 3620, 3640, 3800
c5200, c5300	Cisco AS5200, AS5300
c4000, c4500, c4700	Cisco 4000 series routers
c5fsfc	Catalyst 5000 series
c5rsm	Catalyst 5000 RSM
c5atm	Catalyst ATM
c6400s, c6400r	Cisco 6400 NSP, Cisco 6400 NRP
c6msm	Catalyst
c7000	Cisco 7000, 7010
c7200	Cisco 7200
lgs	IGS, 2500, 3000, and 5100 series routers
gs3	AGS and AGS+ gateway routers
gs7	7000 series gateway routers
Gsr	Gigabit Switch Router (12000)
ls1010	LightStream 1010
Mc3810	Multiservice Cisco 3810
Rpm	MGX 8850
RSP	Cisco 7500 series

*Tabla 4. Identificadores de plataformas*

Grupo de Características: Este grupo representa las funcionalidades que un router con una IOS puede realizar, dichas funcionalidades pueden combinarse pero siempre dependen de que la plataforma pueda soportarlas. La tabla 5 muestra las funcionalidades más comunes.

Grupo	Funcionalidades
A	APPN
A2	ATM
B	AppleTalk
BOOT	Boot image
C	Remote Access Server
D	Desktop subset (SNMP, IP, BRIDGING, WAN, Terminal Services, IPX, ATALK, ARAP)
D2	Desktop Plus
Dsv	Desktop/IBM
EBOOT	Ethernet boot image for MC3810
F	FRAD subset
G	ISDN subset
G2	Gatekeeper proxy, voice and video
G3	ISDN subset for c800 (IP, ISDN, Frame Relay)
I	IP subset
I2	IP subset for 3600
I3	Reduced IP subset with BGP/EGP/NHRP removed
Is	IP Plus
Is56i	IP IPsec 56
J	Enterprise subset (protocol translation)
Js40	Enterprise 40
Jsv56	Enterprise IPsec 56
Js	Enterprise Plus
Js40	Enterprise Plus 40
Js56i	Enterprise Plus IPsec 56
K	Kitchen sink
K1	Privacy key encryption (11.3 and above)
K3	Triple DES (11.3 and above)
K4	168-bit encryption with SSH
L	IpeXchange IPX, static routing, gateway
M	RMON
N	IPX
O	Firewall
O2	Firewall (3xx0)
O3	Firewall with SSH (36x0, 26x0)
P	Service provider (IP RIP/IGRP/EIGRP/OSPF/BGP CLNS ISIS/IGRP)
R	RSRB (remote source route bridging); this option can be added to other feature sets
S	Source route switch (SNMP, IP, BRIDGING, SRB)
V	VIP and dual RSP support
V2	Voice V2D
V3	Voice feature card
X	X.25 (11.1 and earlier), and on 12.0T on c800 series
X	Frame Relay in 11.2
X	H.323 Gatekeeper/Proxy for 2500, 3620, 3640, MC3810
Y	Reduced IP (SNMP, IP RIP/IGRP/EIGRP, BRIDGING, ISDN, PPP), for low-end routers
Y2	IP variant (SNMP, IP RIP/IGRP/EIGRP, WAN -X.25, OSPF, PIM)
Y3	IP/X.31
Y4	Reduced IP variant (Cable, MIBs, DHCP, EZHTTP)
Y5	Reduced IP variant (Cable, MIBs, DHCP, EZIP) for home office
Y6	Reduced IP variant (c800)

Tabla 5. Funcionalidades



La tabla 6 muestra algunos ejemplos de nombres de imágenes CISCO IOS que usan ésta nueva convención:

Nombre	Grupo de características
C2600-ipbase-mz	IP Base
c2600-ipvoice-mz	IP Voice
c2600-advsecurity-mz	Advanced Security
c2600-spservices-mz	Service Provider Services
c2600-entbase-mz	Enterprise Base
c2600-advipservices-mz	Advanced IP Services
c2600-entservices-mz	Enterprise Services
c2600-adventerprise-mz	Advanced Enterprise Services

*Tabla 6. Imágenes de Cisco IOS*

Elegir la versión de Cisco IOS Software que desea ejecutar es bueno, siempre y cuando acepten su hardware, contienen las características que se desea y son compatibles con la memoria del router. Estas son algunas de las recomendaciones y directrices generales:

#### ***Formato de publicación***

Las versiones del software Cisco IOS usa el formato AB (c) D donde:

- A, B son números. Se trata del primer número con punto decimal y sin ningún otro sufijo que aparece. Este número nos indica la versión del software CISCO IOS.
- C es la versión de mantenimiento. Un número más alto significa más mantenimiento de correcciones de errores. Cualquier característica, corrección de errores, y soporte de hardware disponible en una versión de mantenimiento en particular están también disponibles en el siguiente.
- D (si existe) es una letra., si está presente, indica que la liberación no es un lanzamiento importante, sino una extensión de un lanzamiento importante. Estas extensiones suelen proporcionar nuevas características y soporte de hardware nuevo.

Las versiones del CISCO IOS constan de números y letras que denotan diferentes características de la misma según como se agrupan.

**Nota:** Versiones anteriores suelen ser más estables que las nuevas, pero también contienen un menor número de características.

### ***Cisco IOS software Image***

La imagen de software IOS de Cisco es ED, LD, GD, o DF:

- ED significa "rápido despliegue". Las primeras versiones de implementación ofrece nueva funcionalidad, plataforma o soporte de la interfaz. La mayoría de los lanzamientos más importantes no contienen versiones ED.
- GD significa "despliegue general." Un importante lanzamiento de Cisco IOS Software llega a la "Implementación general" hito cuando Cisco cree que es adecuado para el despliegue en cualquier lugar en redes de clientes en las características y funcionalidades de la versión se requieren. Criterios para llegar a la "Implementación general" hito se basan en, pero no limitado a, las encuestas de opinión de los clientes de redes de producción y de ensayo, en los comunicados, informes de clientes Ingeniero de errores, y la experiencia de campo informo. Sólo las versiones principales son candidatos para el hito de implementación general.
- LD significa "despliegue limitado." Un gran lanzamiento del software Cisco IOS se dice que es en el "despliegue limitado" fase de su ciclo de vida durante el período comprendido entre su primer envío y el hito de GD.
- DF significa "diferido". versiones del DF no están disponibles para su descarga, debido a los defectos conocidos. Estos no se deben instalar en su router.

CISCO constantemente adapta su tecnología a los nuevos requerimientos de redes empresariales y de proveedores de servicios, en este sentido, cada nueva versión del software cisco IOS optimiza las redes IP y otorga avanzados servicios de red. El paso de una versión de IOS a otra nueva no es rápido, puesto que se requiere de constantes arreglos y mejoras hasta llegar a la estabilidad del software.

### ***Mecanismo de inferencia***

El mecanismo de inferencia decide, interpreta y aplica el conocimiento de la Base de Conocimientos sobre la Base de Hechos con el fin de obtener la solución buscada.

El funcionamiento es el siguiente:

1. Evaluación, el cual se selecciona el conocimiento a emplear de los métodos de CIDR, NAT o IPv.6
2. Comprobación de que las características de los elementos con los que cuenta el lugar.
3. Ejecución, de los datos introducidos en el SE.
4. Comprobación final de los elementos para elegir el método.

### *Desarrollo de la herramienta para el cálculo de superredes*

El desarrollo de una herramienta que calcule superredes, para redes clase A, B o C, puede ser de gran ayuda para sintetizar, economizar y utilizar en redes que no se toman en cuenta por su poca capacidad de nodos.

### *Escenario de uso de la interfaz para el cálculo de superredes*

**Descripción:** Caso de uso de una calculadora de superredes.

**Actores:** Primario. Desarrollador.  
Secundario. Usuario.

**Disparador:** El actor secundario solicita al sistema calcular una superred.

**Precondiciones:** El sistema no ha calculado una superred.

**Pos condiciones:** El sistema calcula una dirección CIDR, rango de direcciones de superred, wildcard, host mínimo, host máximo, broadcast, número máximo de direcciones obtenidas con la superred y proporciona la clase de red y la dirección IP.

**Flujo Principal:**

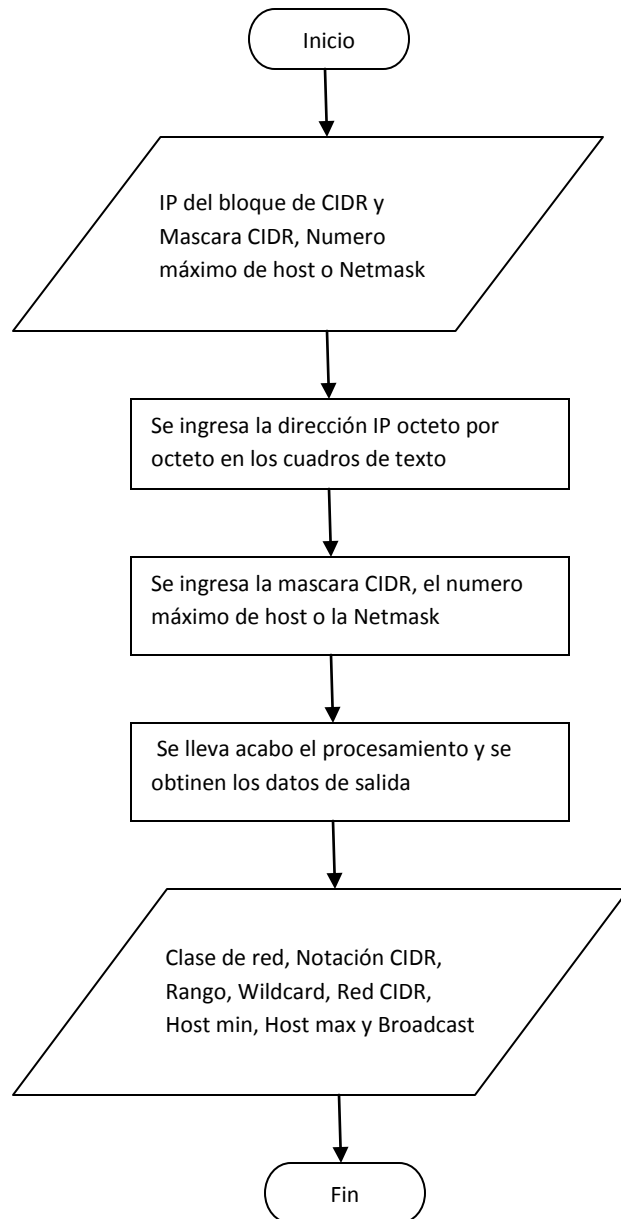
1. El actor secundario inicializa la interfaz de la calculadora de superred.
2. El actor secundario ingresa los datos en la interfaz, como dirección IP de la red y la máscara de red CIDR, número de direcciones requeridas o el número de bits de la máscara.
3. El sistema muestra la dirección CIDR, el rango de direcciones de superred, la wildcard, el host mínimo, el host máximo, el broadcast y la clase a la que pertenece la dirección IP.
4. El actor secundario cierra la interfaz.

**Flujo Secundario:**

2. El actor secundario deja un formulario de la dirección IP vacío.
  - 2.1. La interfaz coloca un cero para llenar los formularios faltantes.
  - 2.2. El actor secundario sustituye el/los cero(s) faltante(s) y el flujo continúa en el punto 3.

**Requerimientos no funcionales:** El proceso para calcular una superred no debe tomar más de 5 minutos por usuario.

*Diagrama de flujo del funcionamiento de la calculadora de superredes*



*Diagrama 2. Diagrama de flujo de funcionamiento de calculadora CIDR.*

### *Escenario de uso del sistema experto*

**Descripción:** Caso de uso de un sistema experto que permite dar una solución al problema de agotamiento de direcciones lógicas IPv.4.

**Actores:** Primario. Desarrollador.  
Secundario. Usuario.

**Disparador:** El actor secundario solicita al sistema una solución al problema de agotamiento de direcciones lógicas IPv.4, para una red LAN clase “A”, “B” o “C” que no cuenta con más direcciones lógicas para asignar.

**Precondiciones:** El sistema experto no ha proporcionado el nombre de un método (IPv.6, NAT o CIDR) como posible solución al problema de agotamiento de direcciones lógicas IPv.4

**Pos condiciones:** El sistema experto proporciona el nombre de un método (IPv.6, NAT o CIDR) como una solución al problema de agotamiento de direcciones lógicas IPv.4 y el procedimiento para llevar a cabo el diagnóstico.

**Flujo Principal:**

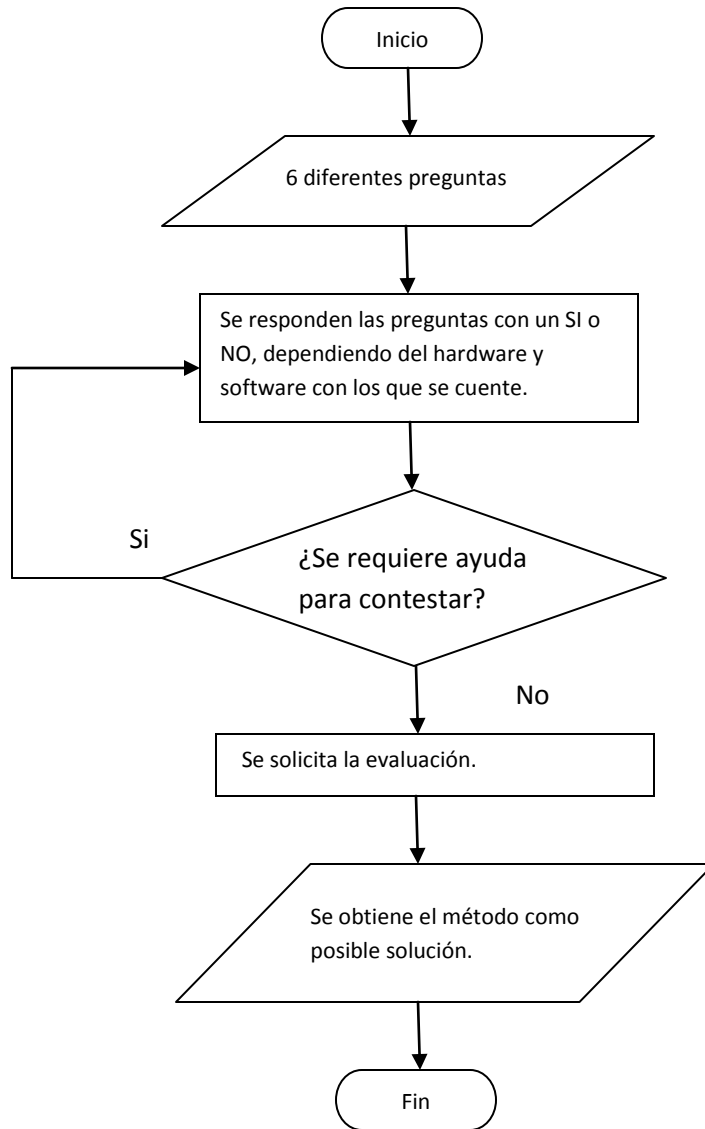
1. El actor secundario inicializa la interfaz del sistema experto.
2. El actor secundario responde a las preguntas mostradas en pantalla, las cuales hacen referencia a la existencia de hardware y software.
3. El sistema experto muestra en pantalla el nombre del método (IPv.6, NAT o CIDR) como una solución al problema de agotamiento de direcciones lógicas IPv.4.
4. El actor secundario cierra la ventana.

**Flujo Secundario:**

2. El actor secundario deja de responder una pregunta en la interfaz y solicita la evaluación de los datos.
  - 2.1. La interfaz muestra un mensaje de error, solicitando contestar las preguntas faltantes.
  - 2.2. El actor secundario responde la(s) pregunta(s) faltante(s) y el flujo continua en el punto 3.
3. El actor secundario presiona el botón de ayuda para ver el procedimiento y poder seguir respondiendo las preguntas.
  - 3.1. El sistema muestra el procedimiento en otra ventana.
  - 3.2. El actor secundario cierra la ventana de procedimiento y el flujo continua en el punto 3.

**Requerimientos no funcionales:** El proceso obtener una solución de agotamiento de direcciones lógicas IPv.4 no debe tomar más de 5 minutos por usuario.

*Diagrama de flujo del Sistema Experto*



*Diagrama 3. Diagrama de flujo del Sistema Experto.*

## ESPECIFICACIÓN TÉCNICA

### *Diseño de un sistema experto que proporcione una solución al problema de agotamiento de direcciones IP*

Se diseñará un sistema experto, que proporcione la solución a implementar para el problema de agotamiento de direcciones IPv4.

Los elementos que contendrá el sistema experto serán los siguientes:

- Base de conocimiento: Características y herramientas que usan los métodos de IP v.6, NAT y superredes.
- Mecanismo de Inferencia: Utilizará encadenamiento hacia delante.
  - Reglas de Inferencia: La forma de evaluación del conocimiento será determinístico (SI-ENTONCES) y con lógica monótona.

### *Base de conocimiento*

#### *CIDR*

El **resumen de rutas**, también llamado **sumarización o agregación de rutas, supernetting, superredes**, etc. es un proceso que realizan los routers por el cual toman un grupo de direcciones de redes contiguas (bloque CIDR) y las resumen en una sola dirección de red común a todas esas redes.

Los protocolos sin clase **EIGRP, OSPF, RIP v.2, IS-IS y BGP** soportan resumen de rutas.

PROTOCOLO	VERSION DE CISCO IOS
<b>EIGRP</b>	12.x
<b>OSPF</b>	12.x
<b>RIP v.2</b>	11.1
<b>IS-IS</b>	12.x
<b>BGP</b>	12.x

*Tabla 7. Versión de Cisco IOS que soporta el protocolo.*

## NAT

### Configuración de NAT

Comando	Qué hace
Router(config)# <b>ip nat pool</b> name start-ip end-ip {netmask netmask   prefix-length prefix-length}	Definir pool o conjunto de ip disponibles sobre las que se va a realizar NAT.
Router(config)# <b>access-list</b> access-list-number <b>permit source</b> [source-wildcard]	Definir el tráfico al que se va a permitir hacer NAT
Router(config)# <b>ip nat inside source list</b> access-list-numb <b>pool</b> name [overload]	Definir el proceso de NAT de las direcciones inside (internas) modificando la dirección de origen (source)

Tabla 7. Configuración NAT en router Cisco.

El NAT dinámico está diseñado para mapear una dirección IP privada a una dirección pública de entre un pool de direcciones públicas ya establecido.

Es decir, cualquier dirección IP pública de este pool se asigna a un Host de la red interna.

1) Primero definir un pool de direcciones (las direcciones públicas que nos asigne nuestro ISP)

```
Router-Cisco#configure terminal
Router-Cisco(config)# ip nat pool 1 200.42.1.1 200.42.1.10 netmask 255.255.255.0
```

2) Crear una lista de acceso Standard que permita las direcciones internas que se deben traducir.

```
Router-Cisco(config)# access-list 1 permit 172.16.129.0 0 0.0.0.255
```

3) Configurar la NAT dinámico basada en la dirección de origen especificando la lista de acceso definida en el paso anterior.

```
Router-Cisco(config)# ip nat inside source list 1 pool 1
```

4) Especificar la interfaz interna y marcarla como conectada al interior.

```
Router-Cisco(config)# interface ethernet 0
Router-Cisco(config-if)# ip nat inside
Router-Cisco(config-if)# exit
```

5) Especificar la interfaz externa y marcarla como conectada al exterior.

```
Router-Cisco(config)# interface serial 0
Router-Cisco(config)# ip nat outside
```



## Comandos para verificación de la tabla NAT

```
show ip nat translations
show ip nat statistics
debug ip nat
```

## **IP v.6**

Software Cisco IOS compatible con IPv6.

12.0S, 12. x T, 12.2S, 12.2SB, 12.2SE, 12.2SR, 12.2SX, 12.2SY, 12.3, 12.4, 15.0, 15.0SY, y 15.1.  
Los trenes de partida 12.0 (22) S, 12.2 (2) T, 12.2 (14) S, 12,2 (28) SB, 12.2 (25) MAR, 12.2 (33) SRA, 12.2 (17a) SX1, 12.2 (50) SY, 12.3, 12.4, 15.0 (1) M, 15.0 (1) SY, y 15.1 (1) S.

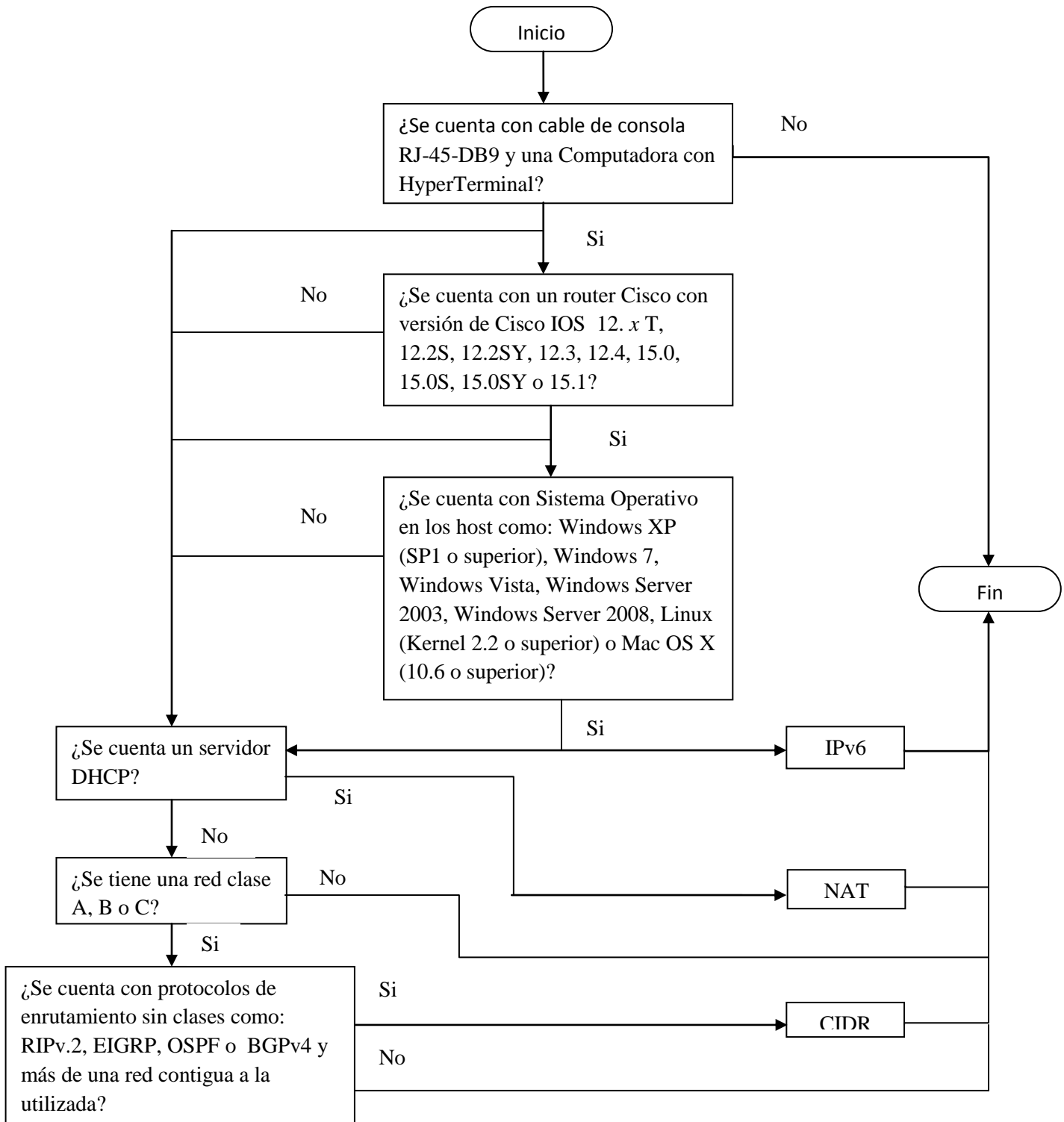
Sistema Operativo compatible con IPv6.

- Windows XP (apartir de SP1).
- Windows Vista.
- Windows 7.
- Windows Server 2003.
- Windows Server 2008.
- Mac OS X 10.6 (Snow Leopard).
- Linux (Desde kernel 2.2).

## **Mecanismo de Inferencia**

- **IF** Se tiene cable de Consola RJ-45-DB9 y una Computadora con HyperTerminal.  
**THEN** Puede configurarse el router
- **IF** Se tiene Protocolos de enrutamiento sin clases como RIPv.2, EIGRP, OSPF o BGPv4 Y más de una red contigua a la utilizada.  
**THEN** Se pueden crear superredes (CIDR).
- **IF** Se cuenta un servidor DHCP **THEN** Puede aplicarse NAT a la Red
- **IF** Se tiene un router Cisco una versión de Cisco IOS 12. x T, 12.2S, 12.2SY, 12.3, 12.4, 15.0, 15.0S, 15.0SY, y 15.1. **THEN** El ruoter es compatible con IPv6.
- **IF** Se cuenta con Sistema Operativo como Windows XP (SP1 o superior), Windows 7, Windows Vista, Windows Server 2003, Windows Server 2008, Linux(Kernel 2.2 o superior),Mac OS X (10.6 o superior). **THEN** Se puede ocupar IPv6.
- **IF** Se tiene una red clase A, B o C .**THEN** Se puede aplicar NAT o CIDR

*Diagrama del Mecanismo de Inferencia*



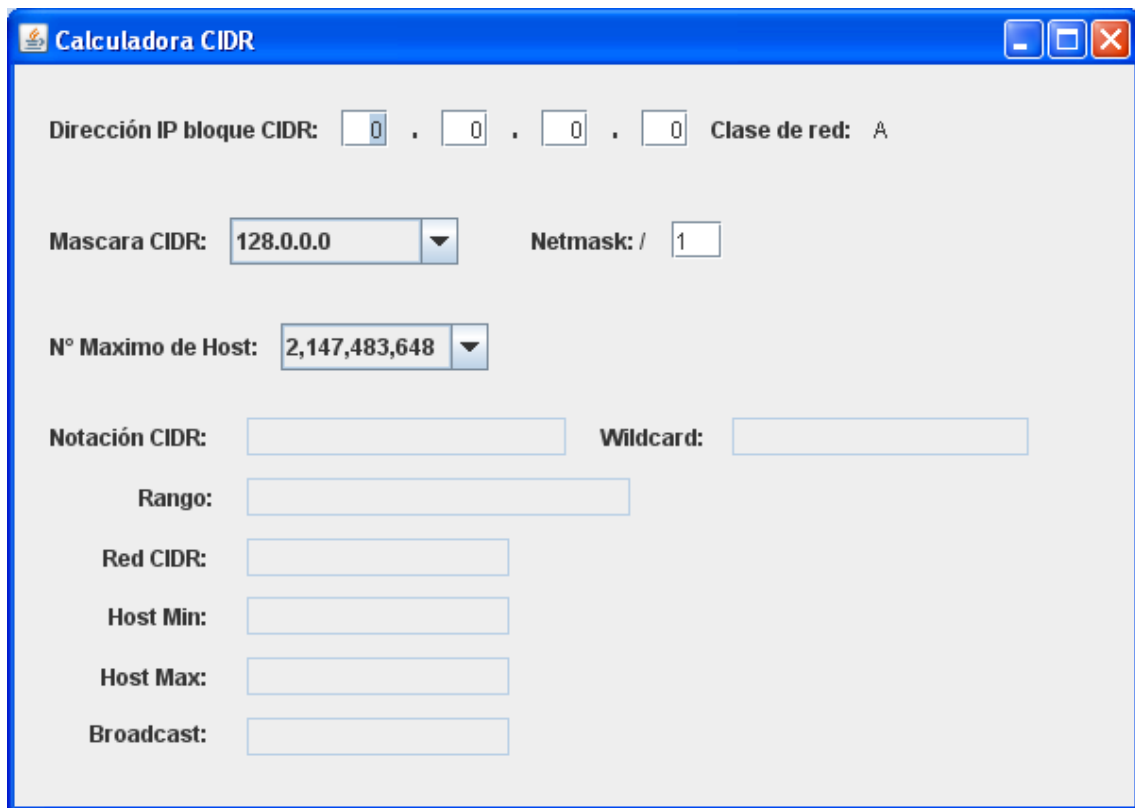
*Diagrama 4. Diagrama de flujo de Mecanismo de Inferencia de SE.*

### *Descripción de la Calculadora de superredes.*

La herramienta calculadora de superredes o CIDR se desarrolló en un ambiente de programación Java con Netbeans versión 6.8 en plataforma Windows XP.

El Funcionamiento es el siguiente:

1. Cuando la aplicación se ejecute, aparece en pantalla, la interfaz grafica en forma de ventana, la cual muestra los formularios a llenar, los datos que se pueden seleccionar y los datos que arrojará después del calculo.



*Imagen 18. Pantalla de Calculadora CIDR.*

2. Los *datos de entrada* o datos a ingresar para realizar el cálculo de superredes son:

- Una dirección IP de red.
- La mascara de red.
- Numero de bits de la mascara de red (Netmask).
- El numero de direcciones que se desean obtener.

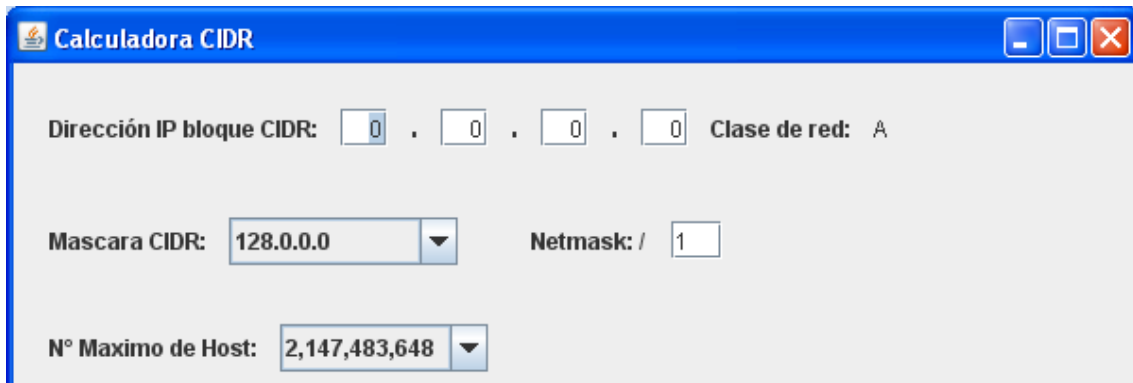


Imagen 19. Datos de entrada de Calculadora CIDR.

3. Cuando el usuario termine de ingresar la dirección IP, se tendrá que elegir la máscara de red, el número de bits de la máscara de red o el número de direcciones que se requieren. Después de hacer la selección de alguno de los tres datos, la aplicación proporcionará los datos de salida. Por ejemplo, si se tiene una máscara de 255.255.252.0 con una dirección de red 10.7.90.8, se tendría rango de direcciones 10.7.88.0 - 10.7.91.255, con 1022 direcciones como máximo para ocupar.

Para poder calcular el rango, se tiene que hacer antes el cálculo de la dirección de red CIDR, que obtiene con la AND de la dirección IP ingresada y la máscara de red. Como se muestra en el siguiente fragmento de código.

```
resred[0]=(Integer.parseInt(sep[0])) & red[0]; //Se recibe el octeto separado de la mascara (en este caso 1er octeto)
resred[1]=(Integer.parseInt(sep[1])) & red[1]; //y se realiza una AND con el octeto de la red (1er octeto). El
resred[2]=(Integer.parseInt(sep[2])) & red[2]; //resultado se almacena en el arreglo resred.
resred[3]=(Integer.parseInt(sep[3])) & red[3];
```

Con la dirección IP resultante, también se obtiene la primera dirección del rango de direcciones, con solo sumarle un 1 al último octeto.

```
//Se forma la cadena para imprimir en pantalla la dirección del host mínimo, donde se suma "1" al último octeto
resred2=resred[0]+ "." +resred[1]+ "." +resred[2]+ "." +(resred[3]+1);
```

Para calcular la Wildcard, se toma la máscara y se le resta 255. Como se muestra en la siguiente imagen del código.

```
//Para el cálculo de la Wildcard, se toma la máscara separada (convirtiendo el dato en entero) y se resta 255
wild[0]=255-(Integer.parseInt(sep1[0]));
wild[1]=255-(Integer.parseInt(sep1[1]));
wild[2]=255-(Integer.parseInt(sep1[2]));
wild[3]=255-(Integer.parseInt(sep1[3]));
```

Con la Wilcard obtenida, se hace el cálculo de la dirección IP máxima a la que llegará el rango (Broadcast).

```
//Para el calculo del rango, se realiza la suma de la wilcard(w) y la red (r), dando como resultado la dirección
//del ultimo host que se obtendra de la red
maxrango[0]=w[0]+r[0];
maxrango[1]=w[1]+r[1];
maxrango[2]=w[2]+r[2];
maxrango[3]=w[3]+r[3];
```

Con la dirección IP máxima del rango o dirección Broadcast, se puede obtener la última dirección que se podrá ocupar del bloque CIDR, con solo restarle un 1 al último octeto.

```
//Se forma la cadena para imprimir en pantalla la dirección del host maximo, donde se resta "1" al ultimo octeto
maxrango2=maxrango[0]+"."+maxrango[1]+"."+maxrango[2]+"."+(maxrango[3]-1);
```

#### 4. Datos de salida:

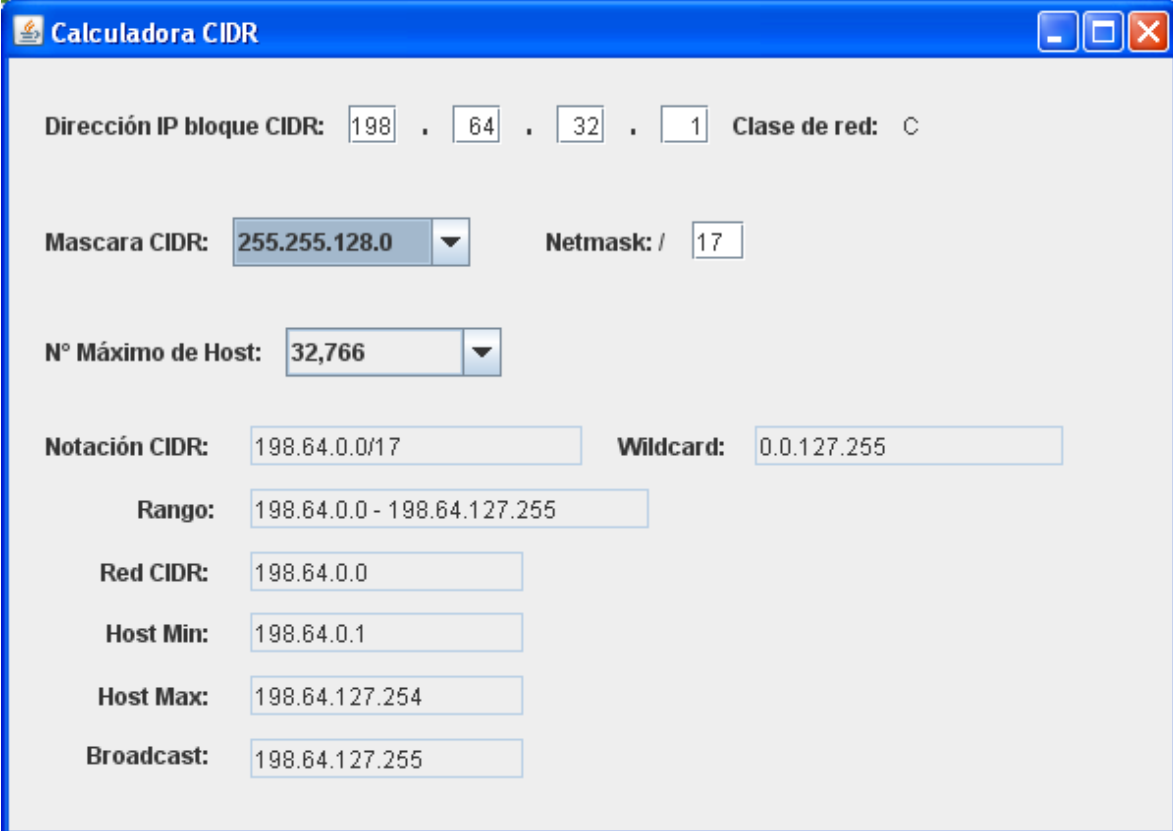
- Dirección de red CIDR.
- Dirección con notación CIDR (ejemplo: 10.17.88.0/21).
- Rango de direcciones de superred.
- Wilcard.
- Broadcast.
- Dirección de mínima de rango.
- Dirección máxima de rango.
- Clase de red.

<b>Notación CIDR:</b>	<input type="text" value="192.0.0.0/3"/>	<b>Wildcard:</b>	<input type="text" value="31.255.255.255"/>
<b>Rango:</b>	<input type="text" value="192.0.0.0 - 223.255.255.255"/>		
<b>Red CIDR:</b>	<input type="text" value="192.0.0.0"/>		
<b>Host Min:</b>	<input type="text" value="192.0.0.1"/>		
<b>Host Max:</b>	<input type="text" value="223.255.255.254"/>		
<b>Broadcast:</b>	<input type="text" value="223.255.255.255"/>		

Imagen 20. Datos de Salida de Calculadora CIDR.

## Ejemplos

**Ejemplo 1.** Dirección IP 198.64.32.1 con máscara CIDR 255.255.128.0.



The screenshot shows a window titled "Calculadora CIDR" with the following fields and values:

Dirección IP bloque CIDR:	198 . 64 . 32 . 1	Clase de red:	C
Máscara CIDR:	255.255.128.0	Netmask: /	17
Nº Máximo de Host:	32,766		
Notación CIDR:	198.64.0.0/17	Wildcard:	0.0.127.255
Rango:	198.64.0.0 - 198.64.127.255		
Red CIDR:	198.64.0.0		
Host Min:	198.64.0.1		
Host Max:	198.64.127.254		
Broadcast:	198.64.127.255		

*Imagen 21. Ejemplo 1.*

**Ejemplo 2.** Dirección IP 64.54.198.3 con Netmask 24.

**Calculadora CIDR**

Dirección IP bloque CIDR:  .  .  .  Clase de red: A

Mascara CIDR:  Netmask: /

Nº Máximo de Host:

Notación CIDR:  Wildcard:

Rango:

Red CIDR:

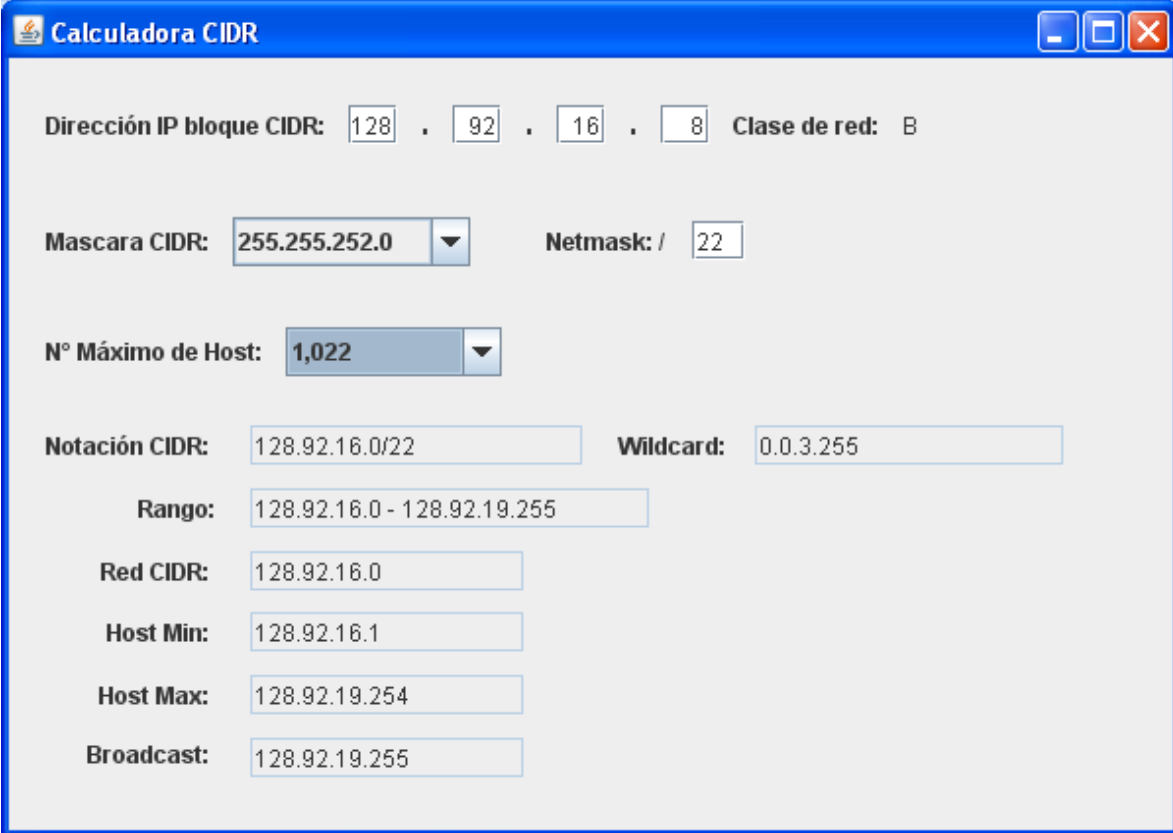
Host Min:

Host Max:

Broadcast:

*Imagen 22. Ejemplo 2.*

**Ejemplo 3.** Dirección IP 128.92.16.8 con 1,022 host.



The image shows a window titled "Calculadora CIDR" with a blue border and standard Windows window controls. The interface is light gray and contains several input fields and labels:

- Dirección IP bloque CIDR:** 128 . 92 . 16 . 8 **Clase de red:** B
- Mascara CIDR:** 255.255.252.0 (with a dropdown arrow) **Netmask: /** 22
- Nº Máximo de Host:** 1,022 (with a dropdown arrow)
- Notación CIDR:** 128.92.16.0/22 **Wildcard:** 0.0.3.255
- Rango:** 128.92.16.0 - 128.92.19.255
- Red CIDR:** 128.92.16.0
- Host Min:** 128.92.16.1
- Host Max:** 128.92.19.254
- Broadcast:** 128.92.19.255

*Imagen 23. Ejemplo 3.*

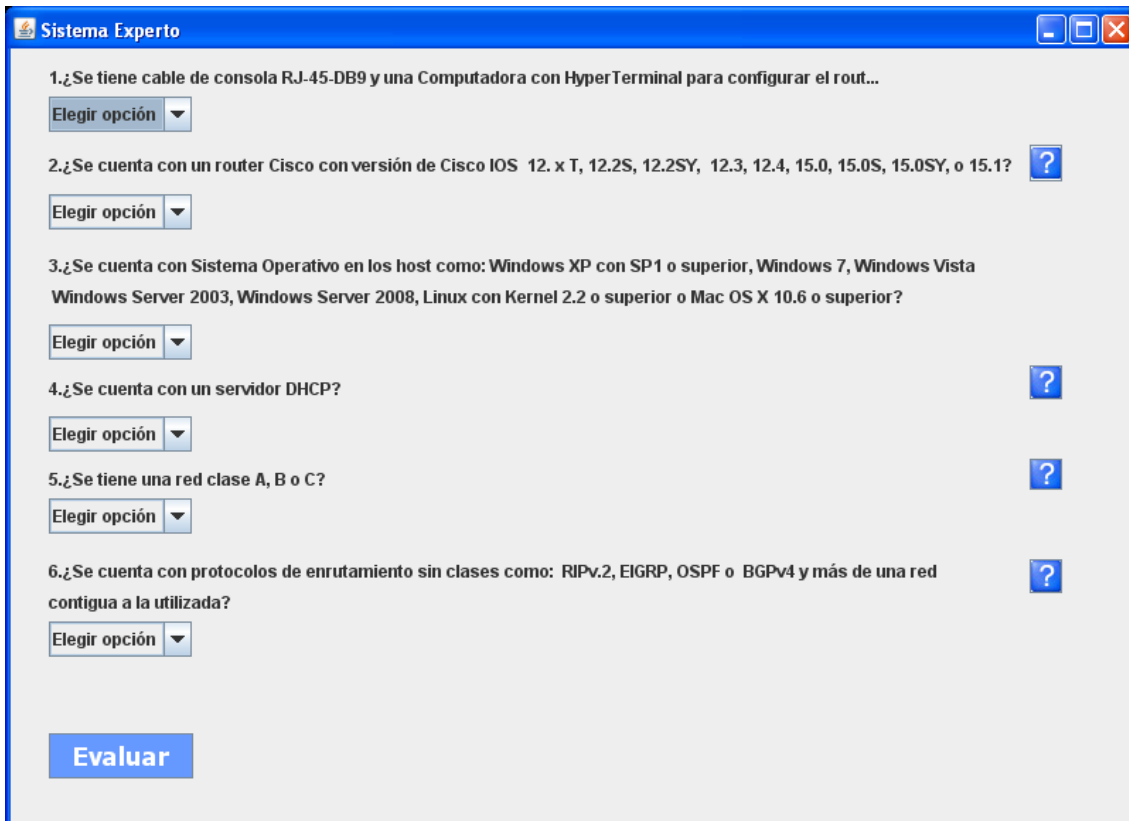


### *Descripción de la interfaz del sistema experto.*

La interfaz se desarrolló en un ambiente de programación java con Netbeans ID 6.8 en plataforma Windows XP.

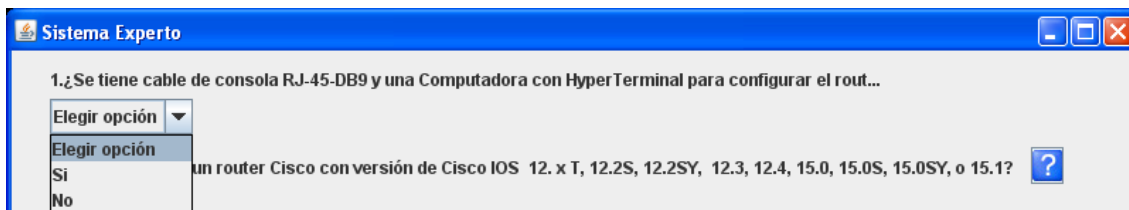
La aplicación cuenta con lo siguiente:

1. Al ejecutar la aplicación se mostrará una ventana que contendrá preguntas sobre la existencia de elementos como hardware y software.



*Imagen 24. Pantalla de Sistema Expero.*

2. El usuario elegirá la respuesta “sí” o “no” de un combo box. Se colocó un combo box debajo de cada pregunta. El usuario contestará conforme a los elementos que se tenga en el lugar.



*Imagen 25. Opciones de respuesta en Sistema Expero.*

3. Al término de contestar las preguntas, se dará clic al botón “Evaluar”. El sistema experto dará como resultado el método que se puede ocupar como solución NAT, CIDR o IPv6. El resultado aparecerá en una ventana emergente.

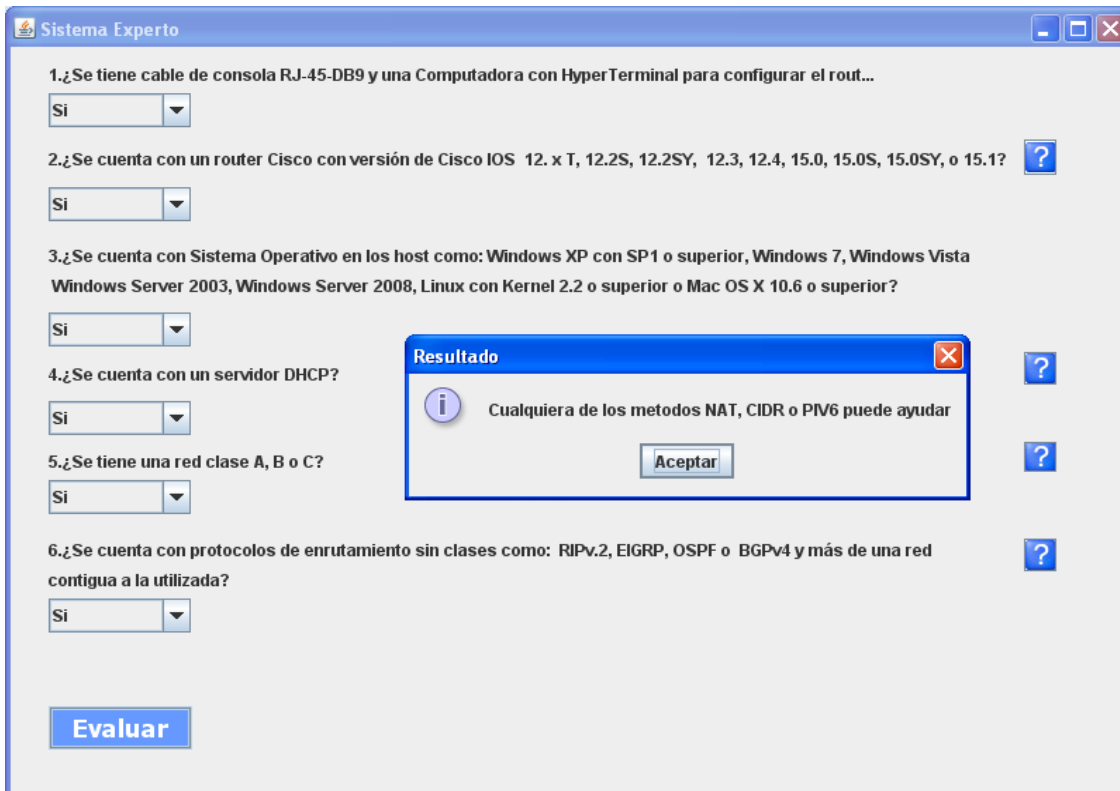


Imagen 26. Resultado de evaluación del Sistema Experto.

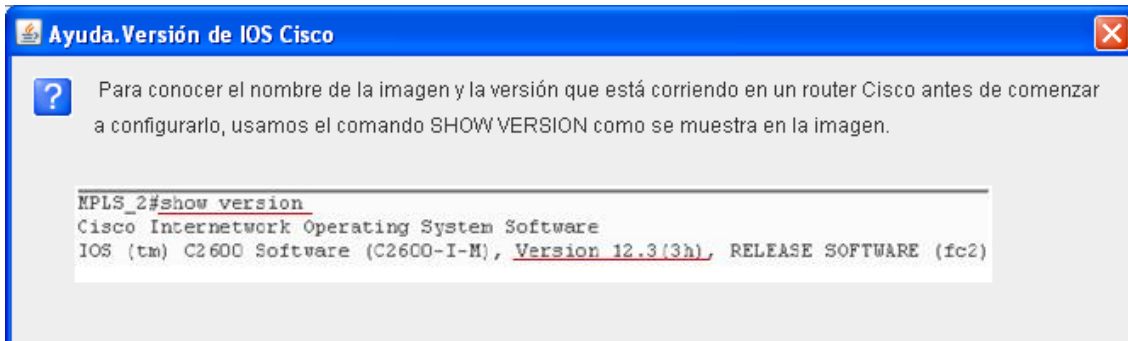
La evaluación de las preguntas se hizo con la sentencia IF-ELSE, como se estipulo en el mecanismo de inferencia.

```

if(res1==1 && res2==1 && res3==1 && res4==1 && res5==1 && res6==1){
    resultado="Cualquiera de los metodos NAT, CIDR o PIV6 puede ayudar";
}
else{
    if(res1==1 && res5==1 && res6==1 && res1==1 && res2==1 && res3==1){
        resultado="El metodo que se puede implementar es: IPv6 y CIDR";
    }
    else{
        if(res1==1 && res2==1 && res3==1 && res1==1 && res4==1){
            resultado="El metodo que se puede implementar es: IPv6 y NAT";
        }
        else{
            if(res1==1 && res5==1 && res6==1 && res1==1 && res4==1){
                resultado="El metodo que se puede implementar es: CIDR y NAT";
            }
            else{
                if(res1==1 && res5==1 && res6==1){
                    resultado="El metodo que se puede implementar es: CIDR";
                }
                else{
                    if(res1==1 && res2==1 && res3==1){
                        resultado="El metodo que se puede implementar es: IPv6";
                    }
                    else{
                        if(res1==1 && res4==1){
                            resultado="El metodo que se puede implementar es: NAT";
                        }
                    }
                }
            }
        }
    }
}

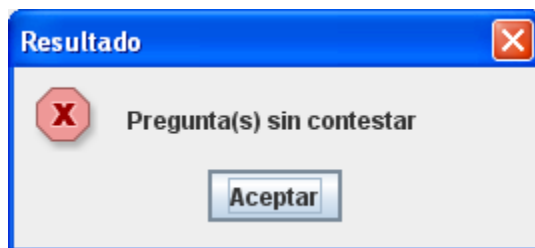
```

En caso de no poder contestar alguna pregunta por desconocimiento de la información, se podrá pulsar el botón de ayuda que se encuentra a un costado de la pregunta. Al oprimirlo se mostrara una ventana con el procedimiento que servirá como una guía para el usuario, porque muestra los pasos a seguir, para obtener la informacion.



*Imagen 27. Pantalla de Ayuda del Sistema Experto.*

En caso de no contestar alguna pregunta, aparecerá un mensaje de error.



*Imagen 28. Pantalla de error del Sistema Experto.*

## Ejemplos

### Ejemplo 1. Evaluacion de preguntas.

The screenshot shows a window titled "Sistema Experto" with the following content:

- Question 1: "¿Se tiene cable de consola RJ-45-DB9 y una Computadora con HyperTerminal para configurar el rout...". Answer: "Si".
- Question 2: "¿Se cuenta con un router Cisco con versión de Cisco IOS 12. x T, 12.2S, 12.2SY, 12.3, 12.4, 15.0, 15.0S, 15.0SY, o 15.1?". Answer: "Si".
- Question 3: "¿Se cuenta con Sistema Operativo en los host como: Windows XP con SP1 o superior, Windows 7, Windows Vista, Windows Server 2003, Windows Server 2008, Linux con Kernel 2.2 o superior o Mac OS X 10.6 o superior?". Answer: "Si".
- Question 4: "¿Se cuenta con un servidor DHCP?". Answer: "No".
- Question 5: "¿Se tiene una red clase A, B o C?". Answer: "No".
- Question 6: "¿Se cuenta con protocolos de enrutamiento sin clases como: RIPv.2, EIGRP, OSPF o BGPv4 y más de una red contigua a la utilizada?". Answer: "No".

A dialog box titled "Resultado" is open, displaying the message: "El método que se puede implementar es: IPv6" with an "Aceptar" button.

At the bottom left of the main window is a button labeled "Evaluar".

Imagen 29. Ejemplo 1.

## Ejemplo 2. Uso de ayuda.

The screenshot shows a window titled "Sistema Experto" with a series of questions and a help dialog box. The questions are:

- ¿Se tiene cable de consola RJ-45-DB9 y una Computadora con HyperTerminal para configurar el rout...  
Si
- ¿Se cuenta con un router...  
Si
- ¿Se cuenta con Sistema Operativo Windows Server 2003, V...  
Si
- ¿Se cuenta con un ser...  
No
- ¿Se tiene una red clas...  
No
- ¿Se cuenta con protocolos de enrutamiento sin clases como: RIPv2, EIGRP, OSPF o BGPv4 y más de una red contigua a la utilizada?  
No

The help dialog box, titled "Ayuda. Clases de red", contains the text: "Para saber que clase de red se tiene hay que considerar lo siguiente:" and a table:

Clase de dirección	Bits de mayor peso	Intervalo de dirección del primer octeto	Número de bits en la dirección de red	Número de redes	Número de hosts por red
Clase A	0	0-127	8	126	16,777,216
Clase B	10	128-191	16	16,384	65,536
Clase C	110	192-223	24	2,097,152	254
Clase D	1110	224-239	28	No es aplicable	No es aplicable

At the bottom of the main window is a blue button labeled "Evaluar".

Imagen 30. Ejemplo 2.

### Ejemplo 3. Mensaje de error al no contestar pregunta.

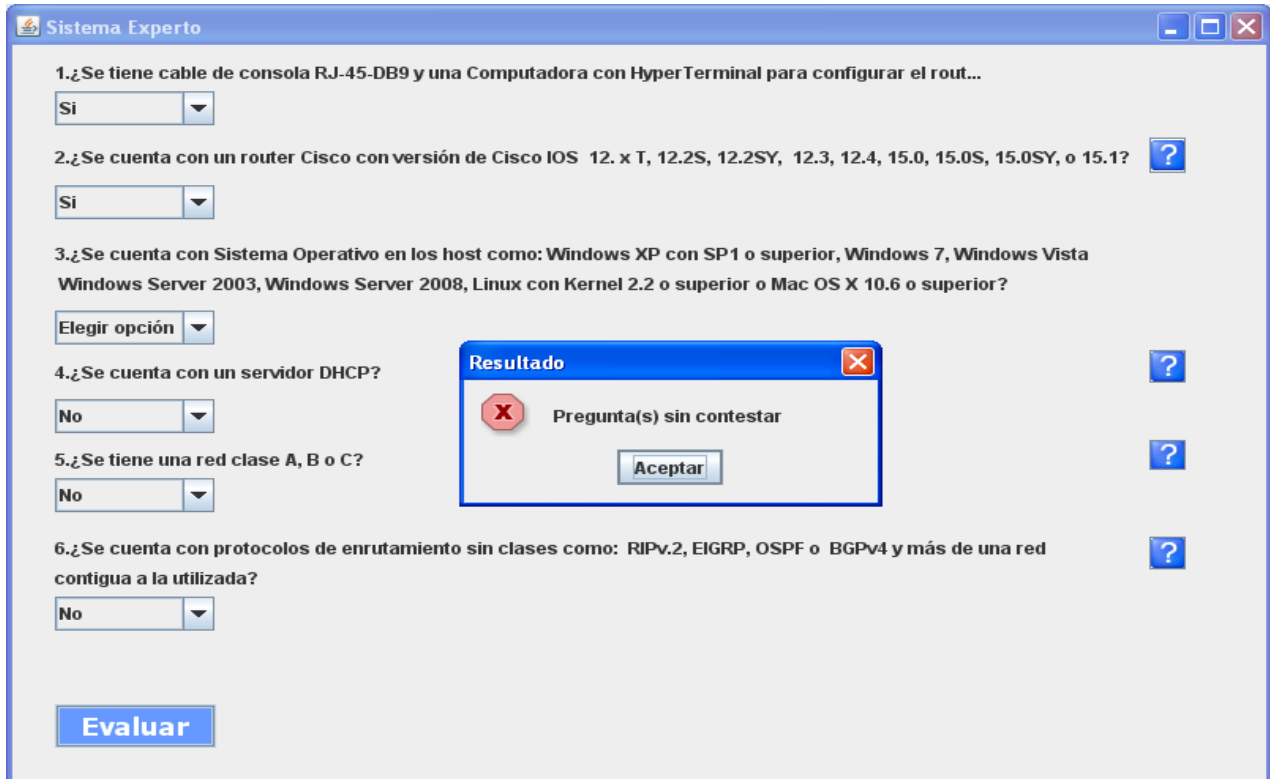


Imagen 31. Ejemplo 3.

### Entregables

Junto con la *documentación* de toda la información acerca del problema a tratar y sus posibles soluciones, se hará entrega de:

- Código fuente de aplicaciones (Anexo 1).
- Archivo ejecutable de las aplicaciones.
- Manual de usuario (Anexo 2).
- Reporte del proyecto en formato PDF.

### Trabajo a futuro

En un futuro, se podrá seguir contando con esta herramienta, ya sea agregando más información o quizá mejorando el ambiente gráfico, así como un sistema experto con mejoras en la interfaz o con mayor información contemplada en la base de conocimiento.

La nueva generación de IP (IPv.6), tardará en colocarse como reemplazo de IPv4, por su alto costo para implementar y porque aún se tiene que seguir desarrollando, por lo que se garantiza el uso de IPv.4 durante varios años más.

## CALENDARIO DE TRABAJO

### Proyecto Terminal de Ingeniería en Computación I Trimestre 12-I

Actividades/Semanas	1	2	3	4	5	6	7	8	9	10	11
Diseñar y desarrollar Base de Conocimiento del sistema experto.											
Diseñar y desarrollar mecanismo de inferencia del sistema											
Implementación del sistema experto a un ambiente gráfico.											

### Proyecto Terminal de Ingeniería en Computación II Trimestre 12-P

Actividades/Semanas	1	2	3	4	5	6	7	8	9	10	11
Diseño de aplicación para cálculo de superredes											
Implementar diseño de aplicación a un ambiente gráfico											

## RECURSOS

Para la elaboración del proyecto, no se contará con equipo de la UAM, ni con herramientas especializadas. El equipo a utilizar será de uso personal:

Se contó con el siguiente hardware:

- Laptop Dell Inspiron 700m
- Disco duro de 40 GB
- Memoria RAM de 1 G
- Procesador Intel Pentium M de 1.60GHz

Se contó con el siguiente software:

- Microsoft Windows XP con licencia
- Microsoft Office 2007 con licencia

Para la programación e implementación de las aplicaciones se usó:

- Netbeans ID 6.8 software libre, en plataforma Windows XP.

## BIBLIOGRAFÍA

[1] M. M. Rodríguez. R. Alcívar. 2011-06-10. [En línea]. “Análisis y diseño de una reingeniería organizativa de la red del campus de la universidad técnica de Manabí mediante la utilización de ip v6. Y su Implementación en la facultad de ciencias Informáticas en el laboratorio de redes”. Disponible: <http://repositorio.utm.edu.ec/bitstream/123456789/.../TESIS%20IPV6.pdf>

[2] Krischan Jodies. 2011-06-20. [En línea]. “IPcalc”. Disponible: <http://www.linux-es.org/ipcalc>.

[3] PocketGear, Inc. 2011-06-21[En línea]. ”CIDR Network Calculator”. Disponible:<http://store.handango.com/es/PlatformProductDetail.jsp?siteId=299&language=spanish&platformId=2&productType=2&productId=274037>.

[4] Santoyo Nelly. 2011-11-3 [En línea]. “Diseño e implementación de un Sistema Experto de Software para brindar información turística en el Estado de Guanajuato”. Disponible: [http://posgrado.itleon.edu.mx/Buscador%20Tesis\\_TEST/Tesis/Sistemas/t-002.pdf](http://posgrado.itleon.edu.mx/Buscador%20Tesis_TEST/Tesis/Sistemas/t-002.pdf).

[5] Cisco Systems, Inc. 2012-03-13 [En línea]. “CCDA Self-Study: RIP, IGRP, and EIGRP Characteristics and Design”. Disponible:<http://www.ciscopress.com/articles/article.asp?p=102174&seqNum=4>

[6] Cisco Systems, Inc. 2012-03-18 [En línea]. “OSPF Design Guide”. Disponible: [http://www.cisco.com/en/US/tech/tk365/technologies\\_white\\_paper09186a0080094e9e.shtml](http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml)

[7] Cisco Systems, Inc. 2012-03-20 [En línea]. “Configuring BGP”. Disponible: [http://www.cisco.com/en/US/docs/ios/12\\_2/ip/configuration/guide/1cfbgp.html](http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfbgp.html)



[8] Cisco Systems, Inc. 2012-03-22 [En línea]. “NAT: Local and Global Definitions”. Disponible: [http://www.cisco.com/en/US/tech/tk648/tk361/technologies\\_tech\\_note09186a0080094837.shtml](http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a0080094837.shtml)

[9] Barrios Joel. 2012-03-20 [En línea]. “Cómo configurar un servidor DHCP en una LAN”. Disponible: <http://www.alcancelibre.org/staticpages/index.php/como-dhcp-lan>

[10] Cisco Systems, Inc. 2012-03-23[En línea]. “How to Choose a Cisco IOS Software Release”. Disponible:[http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products\\_tech\\_note09186a00800fb9d9.shtml](http://www.cisco.com/en/US/products/sw/iosswrel/ps1834/products_tech_note09186a00800fb9d9.shtml)

[11] Diaz Lisset. 2012-03-3En línea]. “PROYECTO FINAL DE CARRERA Evaluación de la herramienta GNS3 con conectividad a enrutadores reales”. Disponible: <http://es.scribd.com/doc/75391581/9/Versiones-del-Cisco-IOS>