



INSTITUTO POLITECNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)



REDES DE COMPUTADORAS

NOMBRE DEL ALUMNO:

- SANTOS MÉNDEZ ULISES JESÚS

NOMBRE DEL MAESTRO:

- JUAN JESÚS ALCARAZ TORRES

TAREA:

- CALCULADORA IP

Introducción

Se desarrollo una calculadora IP que será útil para conocer los valores de la máscara de red, wildcard, número de hosts mínimos y máximos, dirección de broadcast, número de redes, número de subredes, esto se hará introduciendo una dirección IP y el valor de bits de la sub-máscara de red y cuantos bits queremos hacerla transitar.

Se desarrollo con el lenguaje de programación JAVA, esta siendo cursada actualmente la unidad de aprendizaje de POO por lo que los conocimientos en este lenguaje no son los más avanzados, no obstante, se obtuvo el resultado esperado.

Marco Teórico

Subneteo de Redes IP

Es la subdivisión de una red en varias subredes. El subneteo permite a los administradores de red, por ejemplo, dividir una red empresarial en varias subredes sin hacerlo público en Internet. Esto se traduce en que el router que establece la conexión entre la red e Internet se especifica como dirección única, aunque puede que haya varios hosts ocultos. Así, el número de hosts que están a disposición del administrador aumenta considerablemente. (Obsérvese Figura 1).

```
Subnets after transition from /24 to /25
Netmask: 255.255.255.128 = 25 11111111.11111111.11111111.1 00000000
Wildcard: 0.0.0.127           00000000.00000000.00000000.0 11111111
1.
Network: 148.204.58.0/25      10010100.11001100.00111010.0 00000000
HostMin: 148.204.58.1        10010100.11001100.00111010.0 00000001
HostMax: 148.204.58.126      10010100.11001100.00111010.0 11111110
Broadcast: 148.204.58.127    10010100.11001100.00111010.0 11111111
Hosts/Net: 126                Class B
2.
Network: 148.204.58.128/25    10010100.11001100.00111010.1 00000000
HostMin: 148.204.58.129      10010100.11001100.00111010.1 00000001
HostMax: 148.204.58.254      10010100.11001100.00111010.1 11111110
Broadcast: 148.204.58.255    10010100.11001100.00111010.1 11111111
Hosts/Net: 126                Class B
Subnets: 2
Hosts: 252
```

Figura 1. Subneteo de una red en 2

Cuando se introdujo el protocolo de Internet, la Internet Engineering Task Force (IETF) estableció las cinco clases de direcciones IP A, B, C, D y E. Cada una de estas clases puede identificarse por medio del rango de direcciones en el que se encuentran. (Obsérvese Figura 2).

CLASE	DIRECCIONES DISPONIBLES		CANTIDAD DE REDES	CANTIDAD DE HOSTS	APLICACIÓN
	DESDE	HASTA			
A	0.0.0.0	127.255.255.255	128*	16.777.214	Redes grandes
B	128.0.0.0	191.255.255.255	16.384	65.534	Redes medianas
C	192.0.0.0	223.255.255.255	2.097.152	254	Redes pequeñas
D	224.0.0.0	239.255.255.255	no aplica	no aplica	Multicast
E	240.0.0.0	255.255.255.255	no aplica	no aplica	Investigación

* El intervalo **127.0.0.0** a **127.255.255.255** está reservado como dirección loopback y no se utiliza.

Figura 2. Tabla de Clases IP

La clase determina el número de direcciones de red que están disponibles y la cantidad de hosts que albergan las respectivas redes. En la clase A, el primer bloque numérico (también denominado octeto porque un bloque está compuesto por 8 bits) está reservado para la dirección de red y los tres últimos están disponibles para los ID de los hosts, lo que significa que hay pocas redes, pero muchos hosts. En la clase B, los primeros dos bloques son responsables de los Net ID, lo que da como resultado más redes, pero menos hosts. La clase C solo alberga el último octeto para las direcciones de hosts restantes. Por su parte, los rangos de direcciones de las clases D y E están reservados y no se pueden adjudicar.

Funcionamiento del Subnetting

En el subnetting o subneteo se toman bits del ID del host “prestados” para crear una subred. Con solo un bit se tiene la posibilidad de generar dos subredes, puesto que solo se tiene en cuenta el 0 o el 1. Para un número mayor de subredes se tienen que liberar más bits, de modo que hay menos espacio para direcciones de hosts. Cabe remarcar en este caso que tanto las direcciones IP de una subred como aquellas que no forman parte de ninguna tienen la misma apariencia y los ordenadores tampoco detectan ninguna diferencia, de ahí que se creen las llamadas máscaras de subred. Si se envían paquetes de datos de Internet a la propia red, el router es capaz de decidir mediante esta máscara en qué subred distribuye los datos.

Como ocurre con las direcciones de IPv4, las máscaras de red contienen 32 bits (o 4 bytes) y se depositan en la dirección como una máscara o una plantilla.

Importancia del Subnetting

Las secuencias numéricas, las conversiones binarias y las comparaciones lógicas tienen un efecto disuasorio. Sobre todo, en el contexto de la transición a IPv6 muchos se preguntan si realmente merece la pena. La respuesta es claramente afirmativa. Esto es lo que hace que el subneteo también sea relevante para el futuro:

- **Ampliación del rango de direcciones dentro de una red:** el subnetting permite que el administrador de redes pueda decidir el tamaño que tendrán sus redes.
- **Conexión rápida entre los hosts y las subredes:** los paquetes de datos llegan directamente del emisor al receptor y, en principio, no se transmiten por toda la red a través del router. (Obsérvese Figura 3).
- **Mejor organización lógica de los participantes en la red:** para obtener una visión más completa de los hosts, es conveniente hacer una segmentación de estos por departamentos o en función de criterios locales (edificios y plantas diferentes).
- **Mayor grado de seguridad:** si un participante de la red es víctima de un ataque externo, la amenaza se extiende rápidamente a toda la red. El subneteo permite a los administradores de redes aislar las subredes mucho más fácilmente.

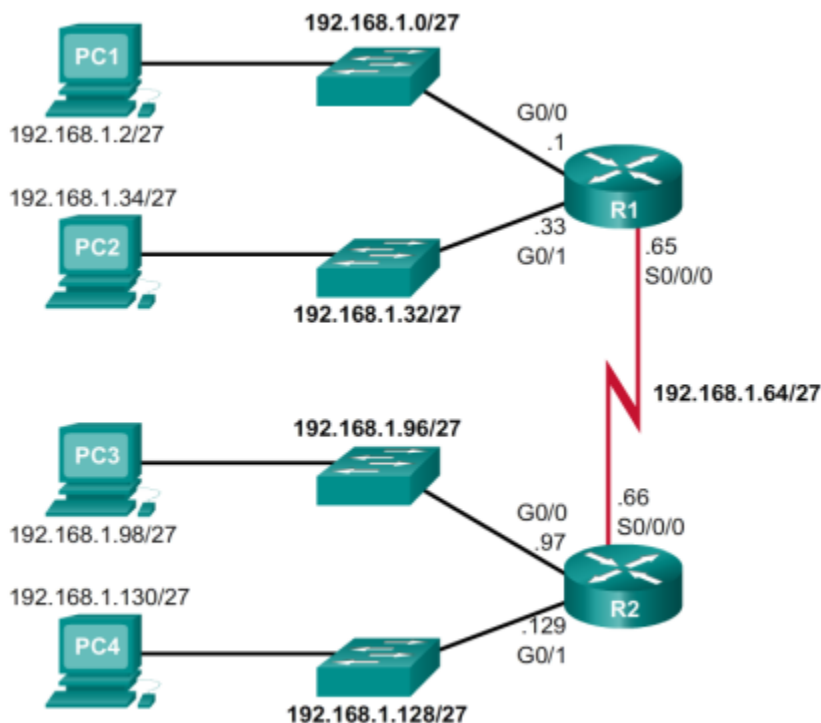


Figura 3. Conexión entre hosts y subredes

Desarrollo

- 1) Se importaron las librerías que se iban a utilizar en la creación y manipulación de la interfaz gráfica.

```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
import java.lang.*;
```

- 2) Se creo un JFrame en JAVA que implementa a ActionListener de manera que nos permita interactuar con los elementos de la interfaz gráfica, el nombre de nuestra clase es Calcip.

```
public class Calcip extends JFrame implements ActionListener {
```

- 3) Se declaran las variables de instancia que serán utilizadas para botones, campos de texto, etiquetas y contadores.

```
private JButton jcalc,jlim;
private JLabel j1,j2,j3,j4,j5,j6,j7,j8,j9,k1,k2,k3,k4,k5,k6,k7,k8;
private JTextField d1,d2,d3,d4,m1,m2,s1,s2,s3,s4,s5,s6,s7,s8,s9,sb1,sb2,sb3;
private int i;
```

- 4) Creación de ventana y panel con sus debidos tamaños, se buscaba que el panel quedará sobre la ventana no obstante esta acción no se pudo lograr, todo esto se encuentra dentro del constructor.

```
//Constructor
public Calcip() {
    JFrame Ventana = new JFrame();
    Ventana.setSize(500,300);
    Ventana.setTitle("CALCULADORA IP");
    Ventana.setLocationRelativeTo(null);
    Ventana.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    Ventana.setVisible(true);
    ImageIcon escudo = new ImageIcon("IPN.JPG");
    Ventana.setIconImage(escudo.getImage());

    JPanel panel = new JPanel();
    panel.setLayout(null);
    panel.setBackground(new Color(0xCCFFCC));
    this.getContentPane().add(panel);
}
```

- 5) Se crean botones, etiquetas y campos de texto, también se inicializan, se posicionan y se agregan al panel.

```
.....  
//Se inicializan y se agregan al panel etiquetas, botones y campos de texto  
j1 = new JLabel("Dirección: ");  
j1.setBounds(400, 10, 70, 20);  
panel.add(j1);  
d1 = new JTextField(3);  
d1.setBounds(460, 10, 30, 20);  
panel.add(d1);  
j2 = new JLabel(".");  
j2.setBounds(495, 10, 30, 20);  
panel.add(j2);  
d2= new JTextField(3);  
d2.setBounds(505, 10, 30, 20);  
panel.add(d2);  
j3 = new JLabel(".");  
j3.setBounds(540, 10, 20, 20);  
panel.add(j3);  
d3= new JTextField(3);  
d3.setBounds(550, 10, 30, 20);  
panel.add(d3);  
  
j4 = new JLabel(".");  
j4.setBounds(585, 10, 20, 20);  
panel.add(j4);  
d4 = new JTextField(3);  
d4.setBounds(595, 10, 30, 20);  
panel.add(d4);  
j5 = new JLabel("Mascara de red(/24)");  
j5.setBounds(640, 10, 120, 20);  
panel.add(j5);  
m1 = new JTextField(2);  
m1.setBounds(760, 10, 30, 20);  
panel.add(m1);  
j6= new JLabel("mover a: ");  
j6.setBounds(800, 10, 80, 20);  
panel.add(j6);  
m2 = new JTextField(2);  
m2.setBounds(860, 10, 30, 20);  
panel.add(m2);  
jcalc = new JButton("Calcular");  
jcalc.setBounds(500, 40, 90, 20);
```

- 6) Se tienen más declaraciones de botones, etiquetas y campos de texto, no obstante, son muchas. Se tiene el método actionPerformed que nos ayudará a realizar operaciones con los JLabel, JTextField y JButton, en este método se llevaron a cabo todas las operaciones desde el cálculo de la máscara de red hasta el cálculo de host por subred.

```
//Metodo para obtener y realizar operaciones con la interaccion de los elementos
public void actionPerformed(ActionEvent e){
```

- 7) Se declaran variables locales que nos permitirán almacenar datos de entrada y ayudarnos a obtener la salida deseada.

```
//Variables locales para el manejo de los dato ingresados
int mod1, mod2, mod3, mod4, tama = 0, rtam = 0, modm1 = 0;
int tmas = 0, var = 0, modm2 = 0, qh = 0, q1, q2, q3, q4;
int icad1, icad2, icad3, exp = 0, val = 0, res = 0;
int smpl = 0, vaux = 0, resta = 0, nsubr = 0;
String part1;
String part2;
String part3;
String bin1 = "";
String bin2 = "";
String bin3 = "";
String bin4 = "";
String bim1 = "";
String f = "0";
```

- 8) Se obtienen las direcciones IP y los bits de la máscara de subred

```
//Obtencion de la direccion IP y del numero de bits de la mascara
JButton jb = (JButton)e.getSource();
int p1 = Integer.parseInt(d1.getText());
int p2 = Integer.parseInt(d2.getText());
int p3 = Integer.parseInt(d3.getText());
int p4 = Integer.parseInt(d4.getText());
int sm1 = Integer.parseInt(m1.getText());
int sm2 = Integer.parseInt(m2.getText());
```

- 9) Se colocaron condicionales para hacer división entre clases de direcciones desde Clase A hasta la Clase E, dentro de la condicional que representa cuando un botón es presionado indicando que todas las operaciones serán realizadas.

- 10) En la primer condicional se tienen cuando el primero octeto es menor a 127

```
if(jb == jcalc){
    if(p1 < 127){//Cuando el primer octeto es menor a 127
        //una clase A
```

- 11) En este segmento se tiene la definición que es una red de clase A, a su vez se tienen las operaciones para obtener subredes y los host por subred.

```
//Se tiene una clase A
j8.setText("Red de Clase A");
s1.setText(""+p1+"."+p2+"."+p3+"."+p4);
q1 = p1;
q2 = p2;
q3 = p3;
resta = sm2 - sm1;
resta = (int) (Math.pow(2, resta));
s8.setText(""+resta);
nsubr = 32 - sm2;
nsubr = (int) (Math.pow(2,nsubr))-2;
s9.setText(""+nsubr);
```

- 12) Se tiene el proceso de conversión del primer y segundo octeto a binario

```
//-----
while(p1>0){//Se obtiene el binario del primer octeto
    mod1 = p1%2;
    bin1 = mod1 + bin1;
    p1 = p1/2;
}
//Se agrega un 0 al MSB del octeto
bin1 = f + bin1;
part1 = bin1;

while(p2>0){//Se obtiene el binario del segundo octeto
    mod2 = p2%2;
    bin2 = mod2 + bin2;
    p2 = p2/2;
}

//Se obtiene el tamaño del octeto y si es menor a 8 se agregan ceros
if(tama < 8){
    f = "";
    tama = bin2.length();
    rtam = 8 - tama;
    for(i=1; i<=rtam;i++){
        f=f+0;
    }
    bin2 = f + bin2;//Se agregan los ceros
}
```



```

    ,
    part2 = bin2;
    while(p3>0){//Se obtiene el binario del tercer octeto
        mod3 = p3%2;
        bin3 = mod3 + bin3;
        p3 = p3/2;
    }
    tama = 0;
    rtam = 0;
//Se obtiene el tamaño del segundo octeto y si es menor a 8 se agregan ceros
    if(tama < 8){
        f="";
        tama = bin3.length();
        rtam = 8 - tama;
        for(i=1; i<=rtam;i++){
            f=f+0;
        }
        bin3 = f + bin3;//Se agregan los ceros
    }
    part3 = bin3;

```

13) Se obtienen los host por red

```

//Se obtienen los hosts por red
    qh = 32 - sml;
    qh = (int) (Math.pow(2, qh)-2);
    s7.setText(""+qh);

```

14) Se obtiene la máscara a partir del valor que se ingresó

```

//Se obtiene la mascara a partir del valor de bits que se ingresa
    if(sml<=8){// se obtiene la mascara
        if(sml == 8){
            s2.setText("255.0.0.0");
            sb2.setText("11111111.00000000.00000000.00000000");
            s3.setText("0.255.255.255");
            sb3.setText("00000000.11111111.11111111.11111111");
            s4.setText(""+q1+".0.0.1");
            s5.setText(""+q1+".255.255.254");
            s6.setText(""+q1+".255.255.255");

```

15) El caso anterior es cuando el valor ingresado es 8 entonces para el se tienen valores fijos de mascara, host mínimo y máximo, broadcast, después se tiene el caso cuando el valor ingresado es menor de 8

```

else{
    sml = 8 - sml;
    smpl = sml;
    sml = (int) (Math.pow(2, 8) - Math.pow(2, sml));
    var = sml;
    vaux = sml;
    s2.setText(""+sml+".0.0.0");
    while(sml>0){//Se obtiene el binario de la mascara
        modml = sml%2;
        bml = modml + bml;
        sml = sml/2;
    }
}

```

```

if(tmas < 8){//Se agregan ceros para formar el octeto
    f = "";
    tmas = bml.length();
    rtam = 8 - tmas;
    for(i=1; i<=rtam;i++){
        f=f+0;
    }
    bml = f + bml;//Se agregan los ceros
}
sb2.setText(""+bml+".00000000.00000000.00000000");
//Se obtiene el Wildcard correspondiente
var = 255 - var;
s3.setText(""+var+".255.255.255");
modml = 0;
bml = "";
while(var>0){//Se obtiene el binario de la mascara
    modml = var%2;
    bml = modml + bml;
    var = var/2;
}

```

16) Para la obtención del host mínimo y máximo se tiene el siguiente algoritmo

```
//-----OBTENCION DE HOST MINIMO, MAXIMO Y BROADCAST-----  
part1 = bin1.substring(smpl+1);  
icad1 = Integer.valueOf(part1);  
while(icad1>0){  
    val = icad1%10;  
    res = res + val*(int)Math.pow(2, exp);  
    exp++;  
    icad1 = icad1/10;  
}  
q1 = q1 - res;  
s4.setText(""+q1+".0.0.1");  
vaux = 255 - vaux;  
vaux = q1 + vaux;  
s5.setText(""+vaux+".255.255.254");  
s6.setText(""+vaux+".255.255.255");  
}  
}  
//-----FINAL DE MASCARA MENOR O IGUAL A 8-----
```

17) Se repite el paso anterior para los valores menores a 16 y menores a 24

18) Cuando se tiene el primer octeto de valor 127 se tiene

```
////////////////////////////////////  
if(p1 == 127){  
    j8.setText("El intervalo 127.x.x.x esta reservado como loopback");  
    s1.setText(""+p1+"."+p2+"."+p3+"."+p4);  
}  
////////////////////////////////////
```

19) Se hace el paso 11 con redes de Clase C

20) Con las redes de clase D y E se colocan valores no definidos ya que no están definidos en la tabla de Subneteo

```
////////////////////////////////////  
if(p1 >= 224 && p1<= 239){  
    j8.setText("Red de Clase D");  
    s1.setText(""+p1+"."+p2+"."+p3+"."+p4);  
    sb1.setText("No disponible");  
    s2.setText("No disponible");  
    sb2.setText("No disponible");  
    s3.setText("No disponible");  
    sb3.setText("No disponible");  
    s4.setText("No disponible");  
    s5.setText("No disponible");  
    s6.setText("No disponible");  
    s7.setText("No disponible");  
    s8.setText("No disponible");  
    s9.setText("No disponible");  
}  
//*****  
////////////////////////////////////  
-  
////////////////////////////////////  
if(p1 >= 240 && p1<= 255){  
    j8.setText("Red de Clase E");  
    s1.setText(""+p1+"."+p2+"."+p3+"."+p4);  
    s2.setText("No disponible");  
    sb1.setText("No disponible");  
    s2.setText("No disponible");  
    sb2.setText("No disponible");  
    s3.setText("No disponible");  
    sb3.setText("No disponible");  
    s4.setText("No disponible");  
    s5.setText("No disponible");  
    s6.setText("No disponible");  
    s7.setText("No disponible");  
    s8.setText("No disponible");  
    s9.setText("No disponible");  
}  
//*****
```

21) Se tiene el botón limpiar que pone en blanco todos los campos de texto

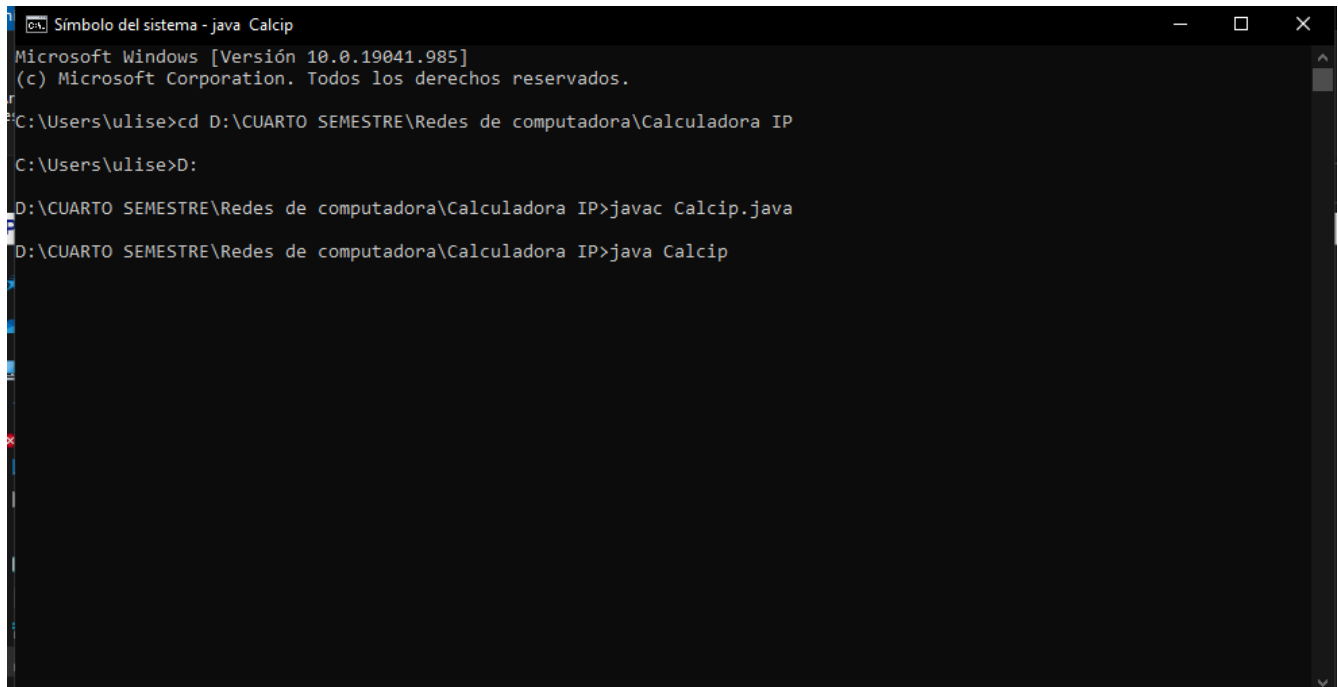
```
//-----BOTON PARA LIMPIAR LOS CAMPOS DE ENTRADA Y SALIDA-----  
  
    if(jb == jlim){  
        j8.setText("");  
        d1.setText("");  
        d2.setText("");  
        d3.setText("");  
        d4.setText("");  
        m1.setText("");  
        m2.setText("");  
        s1.setText("");  
        sb1.setText("");  
        s2.setText("");  
        sb2.setText("");  
        s3.setText("");  
        sb3.setText("");  
        s4.setText("");  
        s5.setText("");  
        s6.setText("");  
        s7.setText("");  
        s8.setText("");  
        s9.setText("");  
    }  
}
```

22) Por último, se tienen el main del programa que nos permitirá ejecutar el programa y visualizar el JFrame

```
//-----MAIN DE LA CALCULADORA-----  
  
    public static void main(String[] args){  
        Calcip calculadora = new Calcip();  
    }  
}
```

Funcionamiento de la calculadora

Se compila y ejecuta el programa de java en la terminal de CMD



```
Símbolo del sistema - java Calcip
Microsoft Windows [Versión 10.0.19041.985]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\ulise>cd D:\CUARTO SEMESTRE\Redes de computadora\Calculadora IP

C:\Users\ulise>D:

D:\CUARTO SEMESTRE\Redes de computadora\Calculadora IP>javac Calcip.java
D:\CUARTO SEMESTRE\Redes de computadora\Calculadora IP>java Calcip
```

Ya que está en ejecución el programa se introducen los valores de los octetos de IP

Dirección: . . .

Se agrega el valor en bits de la mascara de subred y el valor de la transición

Mascara de red(/24) mover a:

Se presiona el botón calcular

Se tiene la siguiente salida de datos

Red de Clase B

Direccion IP:	148.204.58.0	010010100.11001100.00111010.00000000
Mascara:	255.255.0.0	11111111.11111111.00000000.00000000
Wildcard:	0.0.255.255	00000000.00000000.11111111.11111111
Host min:	148.204.0.1	
Host max:	148.204.255.254	
Broadcast:	148.204.255.255	
Hosts/red:	65534	
Subredes:	32	
Host/subred:	2046	

Una vista completa de la interfaz gráfica es

Dirección: 148 . 204 . 58 . 0 Mascara de red(/24) 16 mover a: 21

Calcular Limpiar

Red de Clase B

Direccion IP:	148.204.58.0	010010100.11001100.00111010.00000000
Mascara:	255.255.0.0	11111111.11111111.00000000.00000000
Wildcard:	0.0.255.255	00000000.00000000.11111111.11111111
Host min:	148.204.0.1	
Host max:	148.204.255.254	
Broadcast:	148.204.255.255	
Hosts/red:	65534	
Subredes:	32	
Host/subred:	2046	

Finalmente, se presiona el botón limpiar que nos ayudará a resetear todos los campos de texto

Limpiar



Dirección: . . . Máscara de red(/24) mover a:

Dirección IP:

Máscara:

Wildcard:

Host min:

Host max:

Broadcast:

Hosts/red:

Subredes:

Host/subred:

Referencias:

- <https://www.ionos.mx/digitalguide/servidores/know-how/subnetting-como-funcionan-las-subredes/>

Conclusiones:

En conclusión, se logró realizar la actividad con éxito teniendo el resultado esperado, se tuvieron algunas dificultades como en la conversión a binario de las redes pero finalmente se logró, debido a los pocos conocimientos respecto a POO y Java se hizo un programa bastante largo, que con los conocimientos debidos puedo optimizar el código, reducir variables y reducir líneas de código, a pesar de ser largo el código hace las funciones debidas y realiza las operaciones correctamente.