Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte de Proyecto Terminal

Título de proyecto:
Geo-Localizador mediante Trace Route y cálculo de subredes.

Trimestre:
11-P a 11O

Alumno:
Omar Rodríguez Tapia.

Matricula:
206300337

Asesor:
Rossen Petrov Popnikolov Potchinkov, Profesor titular “C”.
Departamento de Electrónica

1. Introducción

Este reporte describe el proceso de creación de un software de cálculo de subredes, el cual tiene la funcionalidad de proporcionar diversa información sobre una ip y la máscara de subred elegida para el cálculo.

Así como el manejo de un Geo-Localizador de host para la fácil ubicación y la compañía que se dedica a sacar su host a la nube que es el internet.

El objetivo de este proyecto es implementar una aplicación integrada que realice una búsqueda y localización de una computadora en red por medio de un método llamado “trace route”, la cual es visualizada en un mapa del mundo.
También con una dualidad de elección de funciones se puede calcular la subred en la que se localiza algún nodo, permitiendo una mayor información al usuario.

Todo esto con una fácil administración del software, además de ser amigable.

Así como la importancia de las nuevas tecnologías en nuestro mundo cambiante nos hace requerir de aplicaciones que nos hagan más agradable la estancia en nuestra computadora por lo que daré algunas respuestas a los usuarios de esta aplicación, esto para estar conscientes del avance en la tecnología.

2. Descripción del problema

Objetivo general del proyecto

Diseñar e implementar una aplicación integrada para:

1. La Geo-Localización de un nodo mediante Trace Route de manera gráfica, amigable y sencilla.
2. Calculo de subredes.

Objetivos particulares

- Diseñar e implementar software para la resolución de la subred para el nodo buscado.
- Implementar una base de datos de las direcciones IPv4, por estado, ciudad y municipio.
- Diseñar el acceso a un servidor de DNS para traducción de las URLs.
- Implementar un módulo para calcular y desplegar el promedio de tiempo de resultado de consulta.
- Implementar un módulo para la visualización grafica de la localización del nodo buscado, sobre el mapa del mundo.
- Implementar un módulo que permita la captura, análisis y comentarios al tráfico saliente de la estación de pruebas, mediante "Ethereal".
- Realizar pruebas a los diferentes módulos desarrollados.

3. Antecedentes

Proyectos relacionados

Referencias internas.

Se encuentra un proyecto en el que se habla sobre los protocolos de ruteo en

una plataforma cisco [1] pero no es muy parecido a este hablando específicamente sobre una plataforma y centrado en la descripción de protocolos.

No existe dentro de la UAM un proyecto para la geo-localización de computadoras por medio de “trace route”.

Referencias externas.

Adeona [2] es un software libre (OpenDHT) que permite la localización de una computadora mediante envío de información sobre el lugar donde se encuentra, esto por parte de la aplicación instalada en la computadora hacia un servidor en internet para que sea consultada por el dueño, pero en este caso no nos muestra la localización geográfica.

Ip_tracer [3] es una aplicación web que muestra la localización de una computadora pero no es muy dinámico para el cliente, en el sentido de que no ofrece las opciones de obtener la ruta que se traza desde el nodo del usuario hacia algún otro que se busque.

Justificación del proyecto

El aspecto de la localización de una computadora en el mundo es de una importancia muy significativa dado que en muchas empresas recurren a estos métodos para obtener información de alguna computadora.

Hay diversos usos para esta aplicación ya que puede haber ataques informáticos y de esta manera se puede detectar de donde provienen dichos ataques, el amplio uso en que puede trascender es el ámbito de investigación de comunicaciones en general.

Se toma en cuenta la utilidad de una localización de equipos robados con este software, lo cual corresponde con una extrema y potente ayuda para encontrar sus computadoras.

En la actualidad existe software que no es muy presentable o amigable, en estos puntos también va a ser mejorado para aumentar la usabilidad de la aplicación así como la profundización en los detalles de información en la comunicación entre la aplicación y el nodo o computadora encontrada.

Utilidad

La utilidad es amplia:

- Conocimiento de las características inherentes a la ip, como son:
  - La clase de la Ip.
  - Número de subredes que corresponde al cálculo.
  - Número de host que corresponde al cálculo.
  - La subred en la que se encuentra la Ip.
  - El número de host.
  - El Id de la red.
  - Dominio de Broadcast.
  - El rango de Ip asignadas a la subred.
  - Todas las subredes posibles para la combinación de la ip y la máscara de subred.
- Facilitar y comprobar actividades escolares.
- Para una buena elección de tipo de red.
- Entre otros.

4. Desarrollo del proyecto

Fundamentos Teóricos

¿Qué es IP?

Protocolo.

IP es un protocolo no orientado a conexión, usado tanto por el origen como por el destino para la comunicación de datos, a través de una red de paquetes conmutados no fiable y de mejor entrega posible sin garantías.

Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, en IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes.

IP provee un servicio de datagramas no fiable (también llamado del mejor esfuerzo (best effort), lo hará lo mejor posible pero garantizando poco). IP no provee ningún mecanismo para determinar si un paquete alcanza o no su destino y únicamente proporciona seguridad (mediante checksums o sumas de comprobación) de sus cabeceras y no de los datos transmitidos. Por ejemplo, al no garantizar nada sobre la recepción del paquete, éste podría llegar dañado, en otro orden con respecto a otros paquetes, duplicado o simplemente no llegar. Si se necesita fiabilidad, ésta es proporcionada por los protocolos de la capa de transporte, como TCP.

Si la información a transmitir ("datagramas") supera el tamaño máximo "negociado" (MTU) en el tramo de red por el que va a circular podrá ser dividida en paquetes más pequeños, y re ensamblada luego cuando sea necesario. Estos fragmentos podrán ir cada uno por un camino diferente dependiendo de cómo estén de congestionadas las rutas en cada momento.

Las cabeceras IP contienen las direcciones de las máquinas de origen y destino (direcciones IP), direcciones que serán usadas por los enrutadores (routers) para decidir el tramo de red por el que reenviarán los paquetes.

El IP es el elemento común en la Internet de hoy. El actual y más popular protocolo de red es IPv4. IPv6 es el sucesor propuesto de IPv4; poco a poco Internet está agotando las direcciones disponibles por lo que IPv6 utiliza direcciones de fuente y destino de 128 bits (lo cual asigna a cada milímetro cuadrado de la superficie de la Tierra la colosal cifra de 670.000 millones de direcciones IP), muchas más direcciones que las que provee IPv4 con 32 bits. Las versiones de la 0 a la 3 están reservadas o no fueron usadas. La versión 5 fue usada para un protocolo experimental. Otros números han sido asignados, usualmente para protocolos experimentales, pero no han sido muy extendidos.

Internet Protocol (IP)	
Familia:	Familia de protocolos de Internet
Función:	Envío de paquetes de datos tanto a nivel local como a través de redes.
Última versión:	IPv6
Ubicación en la pila de protocolos	
Aplicación	http, ftp, ...
Transporte	TCP, UDP, ...
Red	IP
Enlace	Ethernet, Token Ring, FDDI, ...
Estándares:	RFC 791 (1981) RFC 2460 (IPv6, 1998)

Tabla 0: Protocolo de Internet

Dirección

Una **dirección IP** es una etiqueta numérica que identifica, de manera lógica y jerárquica, a un interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente unacomputadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP. Dicho número no se ha de confundir con la direccion MAC que es un identificador de 48bits para identificar de forma única a la tarjeta de red y no depende del protocolo de conexión utilizado ni de la red. La dirección IP puede cambiar muy a menudo por cambios en la red o porque el dispositivo encargado dentro de la red de asignar las direcciones IP, decida asignar otra IP (por ejemplo, con el protocolo DHCP), a esta forma de asignación de dirección IP se denomina dirección IP dinámica (normalmente abreviado como IP dinámica).

Los sitios de Internet que por su naturaleza necesitan estar permanentemente conectados, generalmente tienen una dirección IP fija (comúnmente, IP fija o IP estática), esta, no cambia con el tiempo. Los servidores de correo, DNS, FTP públicos y servidores de páginas web necesariamente deben contar con una dirección IP fija o estática, ya que de esta forma se permite su localización en la red.

A través de Internet los ordenadores se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, a los seres humanos nos es más cómodo utilizar otra notación más fácil de recordar, como los nombres de dominio; la traducción entre unos y otros se resuelve mediante los servidores de nombres de dominio DNS, que a su vez, facilita el trabajo en caso de cambio de dirección IP, ya que basta con actualizar la información en el servidor DNS y el resto de las personas no se enterarán ya que seguirán accediendo por el nombre de dominio.

IPV4

Las direcciones IPv4 se expresan por un número binario de 32 bits permitiendo un espacio de direcciones de 4.294.967.296 (2^{32}) direcciones posibles. Las direcciones IP se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto está comprendido en el rango de 0 a 255 [el número binario de 8 bits más alto es 11111111 y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255].

En la expresión de direcciones IPv4 en decimal se separa cada octeto por un carácter único ".". Cada uno de estos octetos puede estar comprendido entre 0 y 255, salvo algunas excepciones.

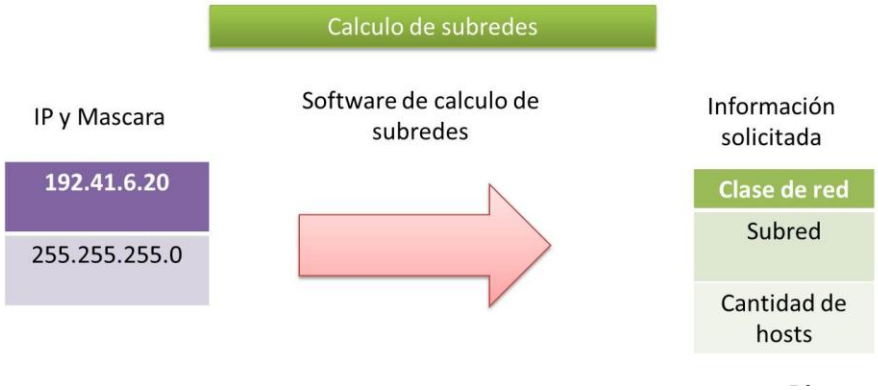
Clase	Rango	N° de Redes	N° de Host Por Red	Máscara de Red	Broadcast ID
A	1.0.0.0 - 127.255.255.255	128	16.777.214	255.0.0.0	x.255.255.255
B	128.0.0.0 - 191.255.255.255	16.384	65.534	255.255.0.0	x.x.255.255
C	192.0.0.0 - 223.255.255.255	2.097.152	254	255.255.255.0	x.x.x.255
(D)	224.0.0.0 - 239.255.255.255	histórico			
(E)	240.0.0.0 - 255.255.255.255	histórico			

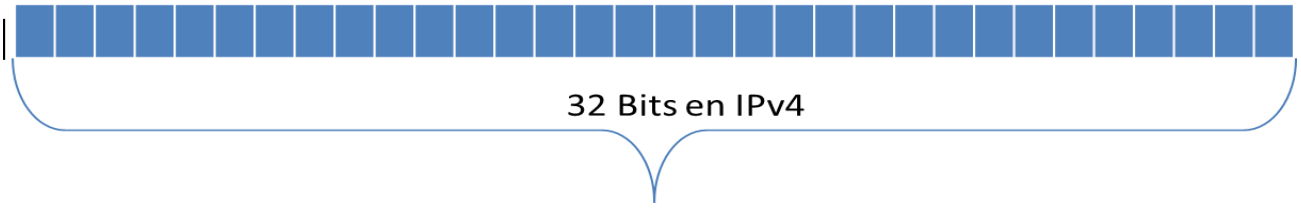
Tabla 1 de Distribución Clases para IPv4

Creación de subredes

El espacio de direcciones de una red puede ser subdividido a su vez creando subredes autónomas separadas. Un ejemplo de uso es cuando necesitamos agrupar todos los empleados pertenecientes a un departamento de una empresa. En este caso crearíamos una subred que englobara las direcciones IP de éstos. Para conseguirlo hay que reservar bits del campo host para identificar la subred estableciendo a uno los bits de red-subred en la máscara. Por ejemplo la dirección 172.16.1.1 con máscara 255.255.255.0 nos indica que los dos primeros octetos identifican la red (por ser una dirección de clase B), el tercer octeto identifica la subred (a 1 los bits en la máscara) y el cuarto identifica el host (a 0 los bits correspondientes dentro de la máscara). Hay dos direcciones de cada subred que quedan reservadas: aquella que identifica la subred (campo host a 0) y la dirección para realizarbroadcast en la subred (todos los bits del campo host en 1).

Calculo de Subredes





Distribución de 32 bits para formar cuatro octetos cada uno con valor máximo de 255.

	Clase A	Clase B	Clase C
Redes validas	1.0.0.0 a 126.0.0.0	128.0.0.0 a 192.0.0.0	192.0.0.0 a 223.255.255.0
Número de redes	(2 ^7) -2	(2 ^14)	(2 ^21)
Numero de host/redes	(2 ^24) -2	(2 ^16) -2	(2 ^8) -2

Tabla 2 #de Redes & Host para IPv4

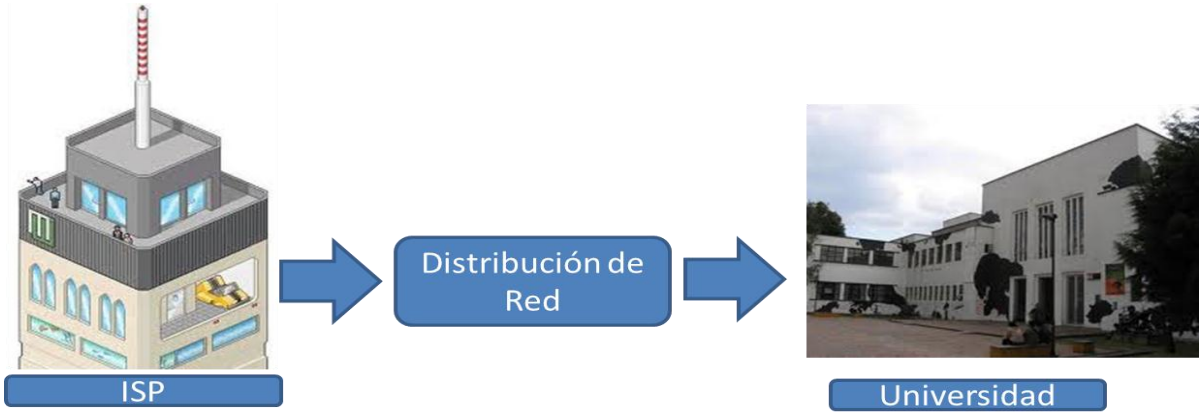
Aquí se decide el uso de la red, que claramente se puede definir por el giro de la empresa que contrata los servicios del ISP, para brindarle el servicio de internet.

Clases

A	0	Red	Host
B	10	Red	Host
C	110	Red	Host

Diagrama 1: Red/Host

Los números de redes son manejados por la corporación ICANN por sus siglas en Inglés (Internet Corporation for Assigned Names and Numbers).
Posteriormente estas direcciones se asignan a, ISP, compañías, universidades, de manera general y por solicitud de necesidades dependiendo de si necesita muchas redes y pocos host o viceversa, o algo intermedio como se observa en el Diagrama 1.



Máscara de subred

La máscara permite distinguir los bits que identifican la red y los que identifican el host de una dirección IP. Dada la dirección de clase A 10.2.1.2 sabemos que pertenece a la red 10.0.0.0 y el host al que se refiere es el 2.1.2 dentro de la misma. La máscara se forma poniendo a 1 los bits que identifican la red y a 0 los bits que identifican el host. De esta forma una dirección de clase A tendrá como máscara 255.0.0.0, una de clase B 255.255.0.0 y una de clase C 255.255.255.0. Los dispositivos de red realizan un AND entre la dirección IP y la máscara para obtener la dirección de red a la que pertenece el host identificado por la dirección IP dada. Por ejemplo un router necesita saber cuál es la red a la que pertenece la dirección IP del datagrama destino para poder consultar la tabla de encaminamiento y poder enviar el datagrama por la interfaz de salida. Para esto se necesita tener cables directos. La máscara también puede ser representada de la siguiente forma 10.2.1.2/8 donde

el /8 indica que los 8 bits más significativos de máscara están destinados a redes, es decir /8 = 255.0.0.0. Análogamente (/16 = 255.255.0.0) y (/24 = 255.255.255.0).

Combinación de bits	Valor Decimal
1 0 0 0 0 0 0 0 →	128
1 1 0 0 0 0 0 0 →	192
1 1 1 0 0 0 0 0 →	224
1 1 1 1 0 0 0 0 →	240
1 1 1 1 1 0 0 0 →	248
1 1 1 1 1 1 0 0 →	252
1 1 1 1 1 1 1 0 →	254
1 1 1 1 1 1 1 1 →	255

Tabla 3: Valores Posibles para una Mascar de Subred (c/octeto)

IP dinámica

Una dirección IP dinámica es una IP asignada mediante un servidor DHCP (Dynamic Host Configuration Protocol) al usuario. La IP que se obtiene tiene una duración máxima determinada. El servidor DHCP provee parámetros de configuración específicos para cada cliente que desee participar en la red IP. Entre estos parámetros se encuentra la dirección IP del cliente.

DHCP apareció como protocolo estándar en octubre de 1993. El estándar RFC 2131 especifica la última definición de DHCP (marzo de 1997). DHCP sustituye al protocolo BOOTP, que es más antiguo. Debido a la compatibilidad retroactiva de DHCP, muy pocas redes continúan usando BOOTP puro.

Las IP dinámicas son las que actualmente ofrecen la mayoría de operadores. El servidor del servicio DHCP puede ser configurado para que renueve las direcciones asignadas cada tiempo determinado.

PING

Formalmente, PING el acrónimo de Packet Internet Groper, el que puede significar "Buscador o rastreador de paquetes en redes".
Como programa, ping es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta. Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.

Ejecutando Ping de solicitud, el Host local envía un mensaje ICMP, incrustado en un paquete IP. El mensaje ICMP de solicitud incluye, además del tipo de mensaje y el código del mismo, un número identificador y una secuencia de números, de 32 bits, que deberán coincidir con el mensaje ICMP de respuesta; además de un espacio opcional para datos.
Muchas veces se utiliza para medir la latencia o tiempo que tardan en comunicarse dos puntos remotos, y por ello, se utiliza el término PING para referirse al lag o latencia de la conexión en los juegos en red.
Existe otro tipo, Ping ATM, que se utiliza en las redes ATM, y en este caso, las tramas que se transmiten son ATM (nivel 2 del modelo OSI). Este tipo de paquetes se envían para probar si los enlaces ATM están correctamente definidos.

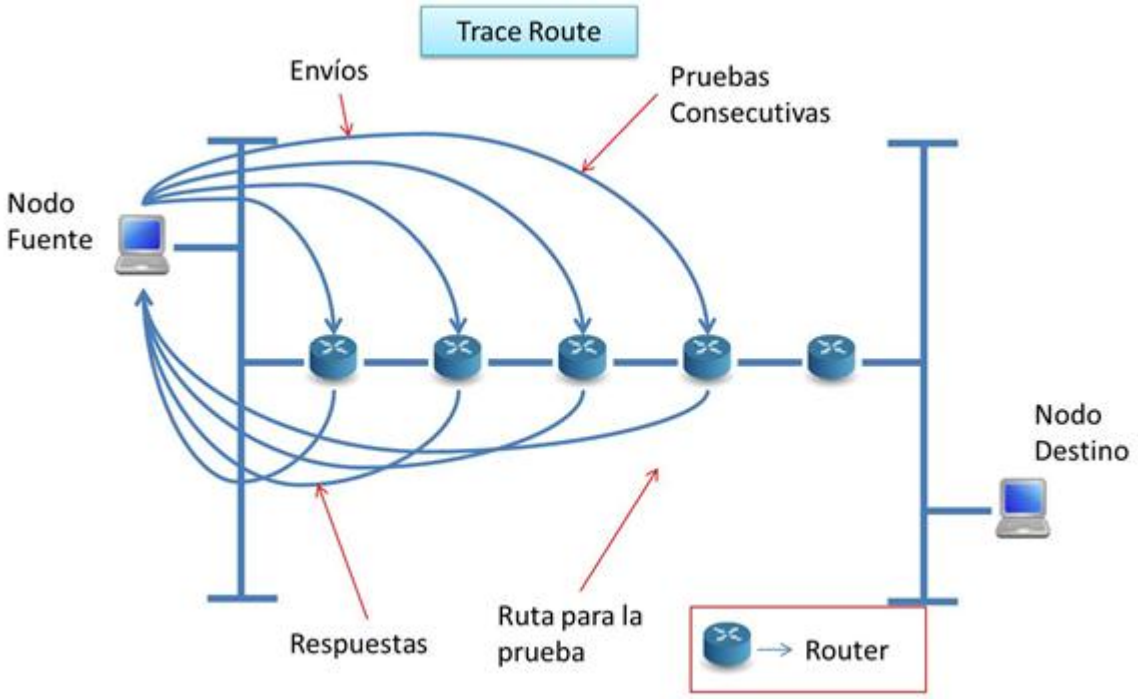
La utilidad Ping trabaja en la capa de red del protocolo TCP/IP y es un tipo de mensaje de control del protocolo ICMP, sub-protocolo de IP. El funcionamiento de Ping y del protocolo ICMP, en general, están definidos en RFC.
El protocolo IP encapsula el mensaje ICMP dentro de un paquete y lo envía. Suele llamarse Paquete ICMP. En el paquete pueden distinguirse dos conjuntos de datos: La Cabecera IP, que contiene los datos estándar de la Capa de red, y el sub-paquete ICMP, que contiene los datos de control. En la Cabecera IP se especifican los valores protocolo como 1 y tipo de servicio como 0 de forma obligatoria. En el sub-paquete ICMP se especifican los valores tipo de mensaje ICMP a 8 (petición) ó 0 (respuesta) y code a 0 (en ambos casos).

El total de la cabecera IP no podrá superar los 160 bits (20 bytes), tras la cual se situará el mensaje ICMP, con un tamaño estándar de 64 bits (8 bytes).

Traceroute

Traceroute es una consola de diagnóstico de redes de Linux que permite seguir la pista de los paquetes que vienen desde un host (punto de red) host. Se obtiene además una estadística del RTT o latencia de red de esos paquetes, lo que viene a ser una estimación de la distancia a la que están los extremos de la comunicación. Esta herramienta se llama traceroute en UNIX y GNU/linux, mientras que en Windows se llama tracert.

Tracert utiliza el campo Time To Live (TTL) de la cabecera IP. Este campo sirve para que un paquete no permanezca en la red de forma indefinida (por ejemplo, debido a la existencia en la red de un bucle cerrado en la ruta). El campo TTL es un número entero que es de-crecerá por cada nodo por el que pasa el paquete. De esta forma, cuando el campo TTL llega al valor 0 ya no se reenviará más, sino que el nodo que lo esté manejando en ese momento lo descartará. Lo que hace tracert es mandar paquetes a la red de forma que el primer paquete lleve un valor TTL=1, el segundo un TTL=2, etc. De esta forma, el primer paquete será eliminado por el primer nodo al que llegue (ya que éste nodo de-crecentará el valor TTL, llegando a cero). Cuando un nodo elimina un paquete, envía al emisor un mensaje de control especial indicando una incidencia. Tracert usa esta respuesta para averiguar la dirección IP del nodo que desechó el paquete, que será el primer nodo de la red. La segunda vez que se manda un paquete, el TTL vale 2, por lo que pasará el primer nodo y llegará al segundo, donde será descartado, devolviendo de nuevo un mensaje de control. Esto se hace de forma sucesiva hasta que el paquete llega a su destino.



Asociación de una Dirección ip con un País.

Se calcula el numero IPN.

IPN

Es el número que se abstrae de la dirección IP con un formato de 4 octetos con máximo 255 (en Decimal) cada uno.

La formula es dado un formato de IP. A.B.C.D:

$$IPN = Ax(256*256*256)+Bx(256*256)+Cx(256)+D$$

Este número se verá asociado a una base de datos que contiene una relación IPN inicio, IPN Fin, País asociado.

DNS

Domain Name System o DNS (en español: sistema de nombres de dominio) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos

equipos mundialmente.
El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.
La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio FTP de prox.mx es 200.64.128.4, la mayoría de la gente llega a este equipo especificando ftp.prox.mx y no la dirección IP. Además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Domain Name System (DNS)	
Familia:	Familia de protocolos de Internet
Función:	Resolución de nombres de dominio
Puertos:	53/UDP, 53/TCP
Ubicación en la pila de protocolos	
Aplicación	DNS
Transporte	TCP o UDP
Red	IP (IPv4, IPv6)
Estándares:	RFC 1034 (1987) RFC 1035 (1987)

Tabla 5: Domain Name System

Diseño

Metodología de base empleada en el proyecto

La metodología utilizada fue una programación orientada a Eventos que provee una fácil interacción con el usuario de la aplicación.

Se utilizó dado que se pretende realizar patrón fachada para poder, hacer transparente para el usuario la complejidad de la programación y la profunda funcionalidad de la aplicación.

Modelado del sistema
Calculadora de Subredes (Subnetting)

Diagrama Jerárquico de clases.

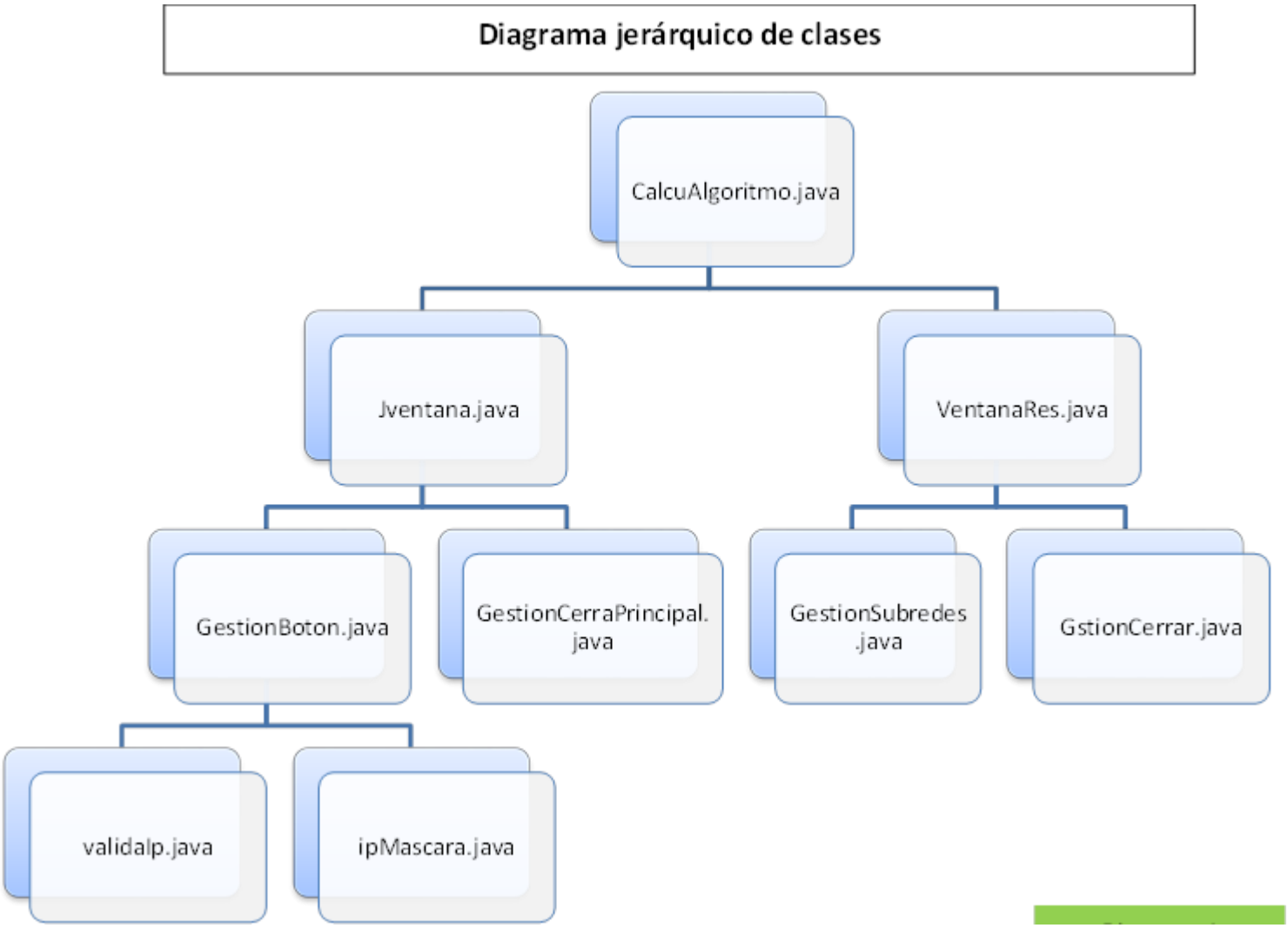
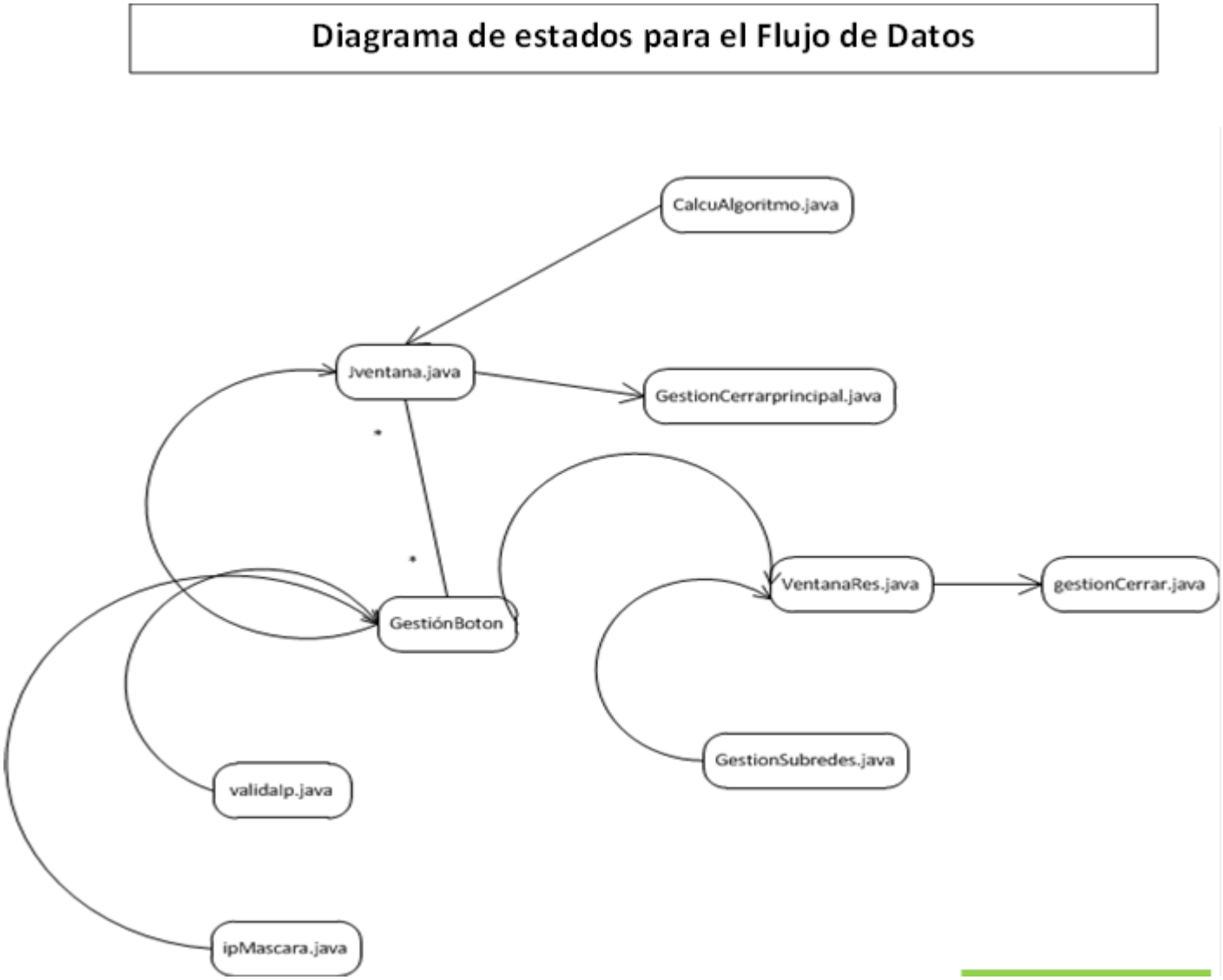


Diagrama de estados para el Flujo de Datos .

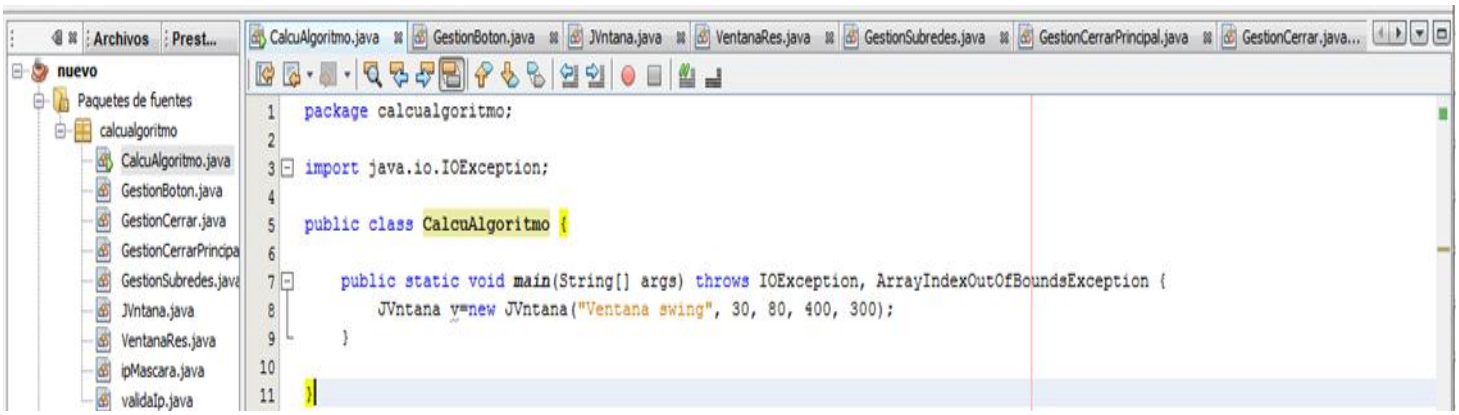


Descripción de clases.

- **CalcuAlgoritmo.java**

Esta es la clase principal, aquí se instancia el método principal, el cual crea la ventana principal para el cálculo de subred.
Esta hace la creación de un objeto Java Swing que recibe los parámetros de

- Titulo
- Ancho
- Alto
- Posición x
- Posición y

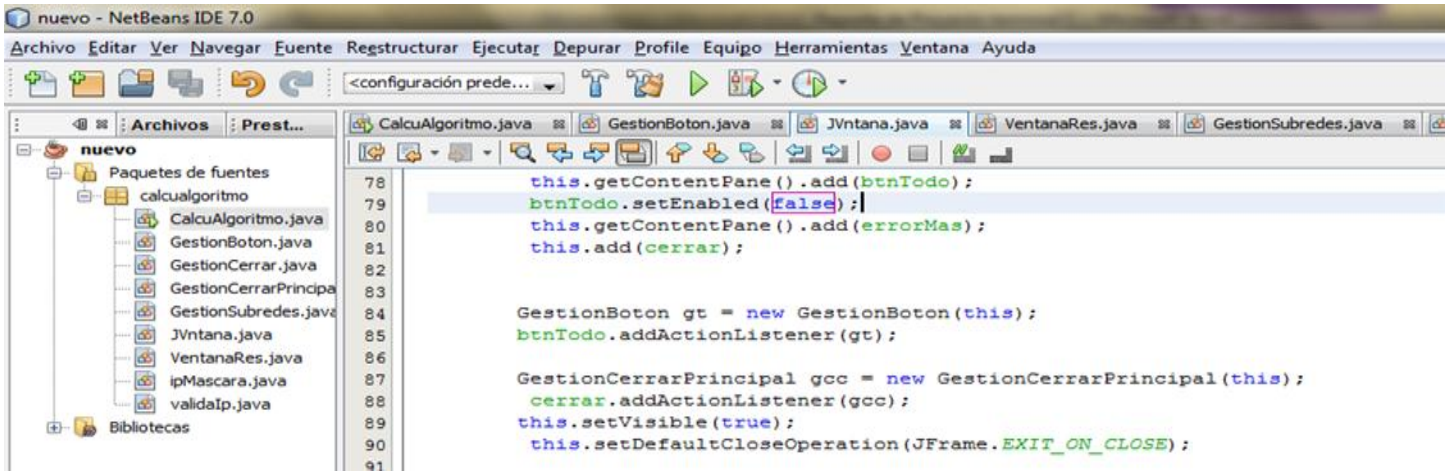


- **Jventana.java**

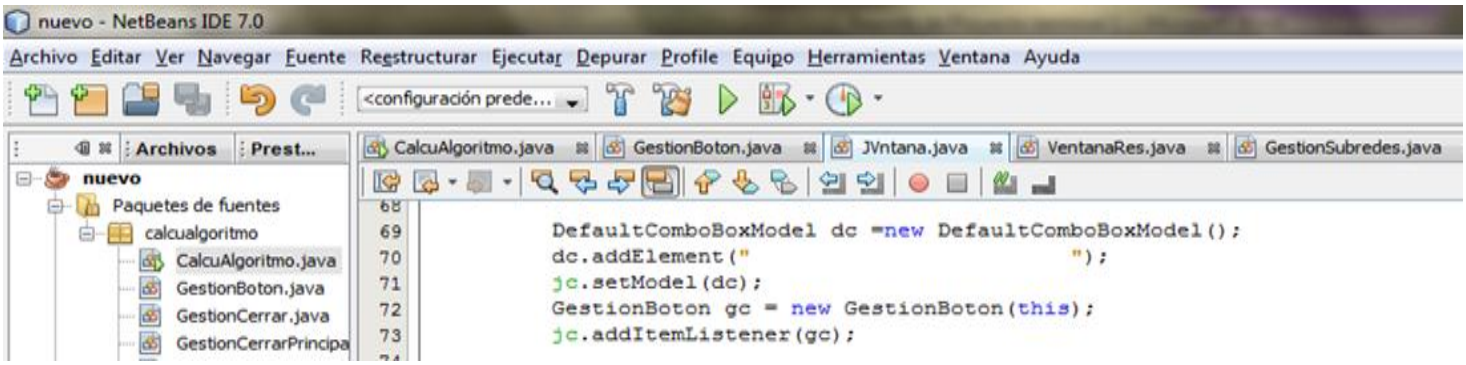
En esta clase se implementa la ventana principal, en la cual se capturarán los datos, los cuales son: la ip y la máscara de Subred, dicha ventana proporciona los botones de “Enviar Ip”, “Enviar Todo” y “Cancelar”.
Aquí coloqué una ip por default la cual es editable para cualquier ip valida.

El botón “Enviar todo” crea un objeto que controla el evento de pulsación de este añadiendo un “action listener”, la clase llamada “GestionBoton.java” es la encargada de manejar dicho evento.

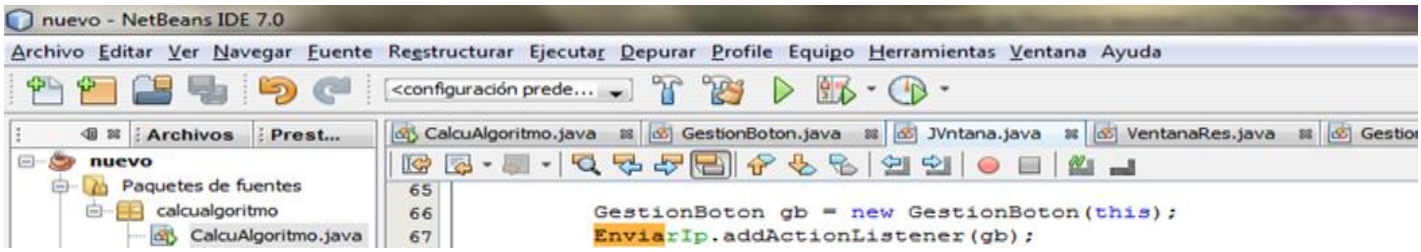
El combo box que muestra las posibles mascaras de subred le añadí un “actionListener” para que en el momento de la selección este evento se capture y tome dicha mascara.



El combo box que muestra las posibles mascaras de subred le añadí un "actionListener" para que en el momento de la selección este evento se capture y tome dicha mascara.

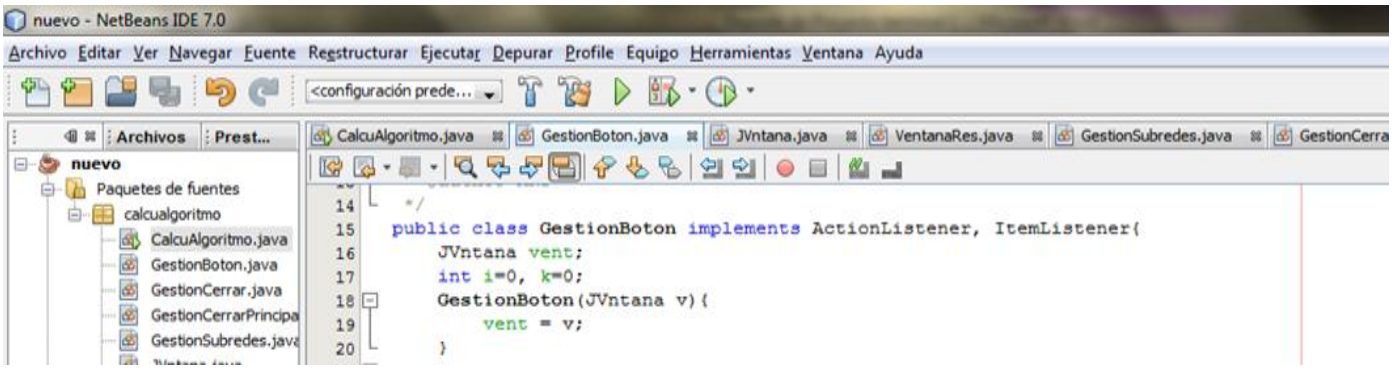


También en esta clase se gestiona el evento del botón "enviar ip", el cual valida la ip escrita en los textField con una adición de un ActionListener.



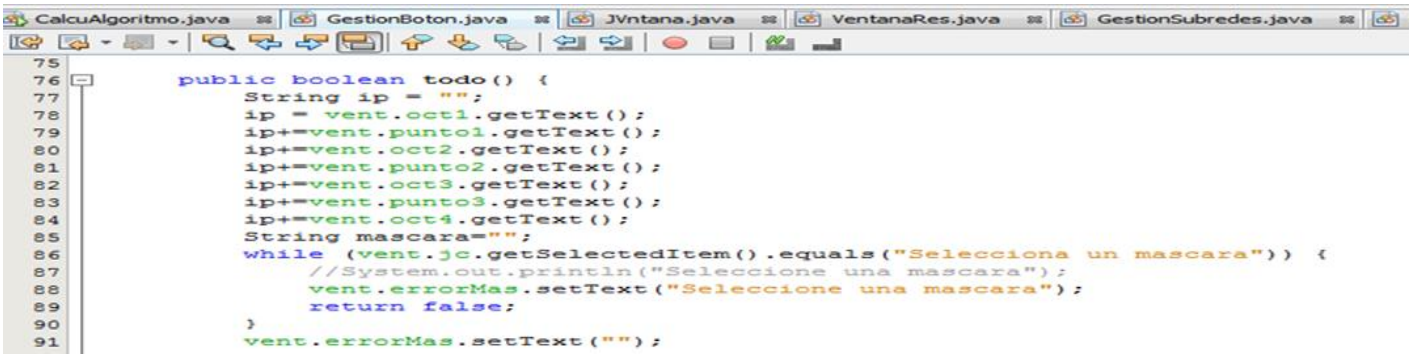
- **GestionBoton.java**

En esta clase se gestiona el evento al dar Click en el botón "Enviar Todo", "Enviar ip", así como el evento de selección de la máscara de subred en el combo box, Implementando de las interfaces "ActionListener" e" ItemListener".



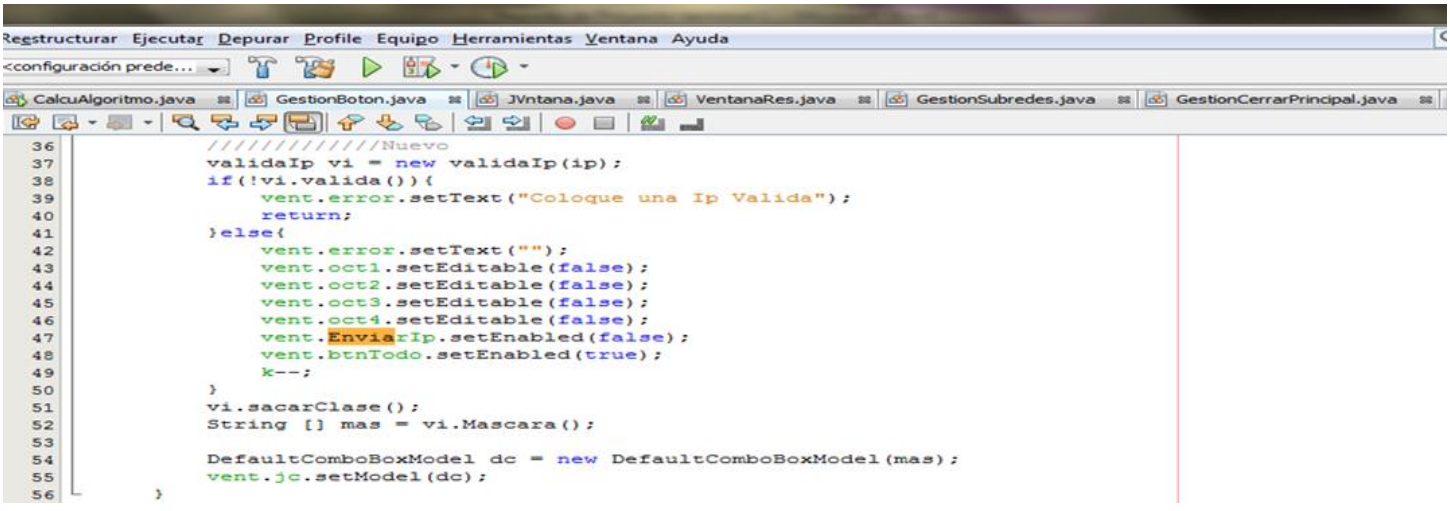
- **Click botón Enviar Todo ActionListener.**

Aquí se utiliza el método llamado "todo" para enviar toda la información recibida de la ventana principal para su respectivo cálculo, seleccionando la ip contenida en 4 texfield y la mascar de subred elegida por el usuario.



- **Click botón Enviar Ip ActionListener.**

Aquí se valida la ip que escribimos creando un nuevo objeto llamado validaIp que contiene los métodos de validación, y a la vez que se presionó el botón, si esta ip es válida se deshabilitan los campos para la entrada de la ip.

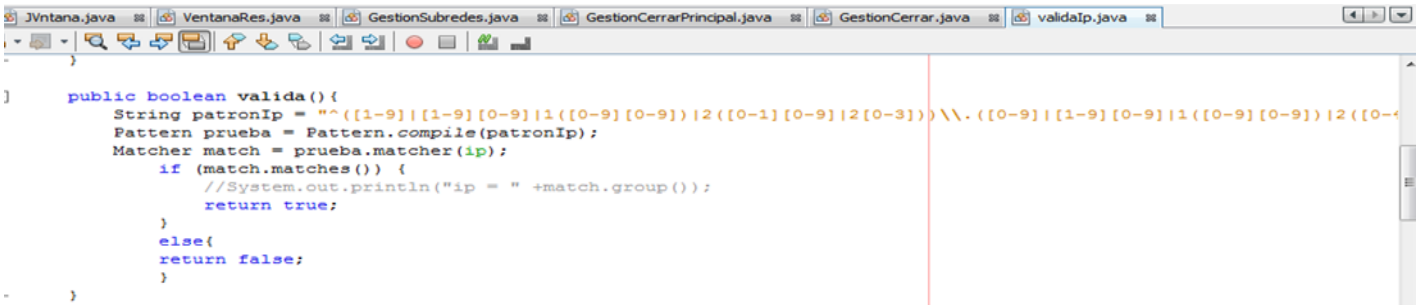


 Selección de Mascara de subred ItemListener.

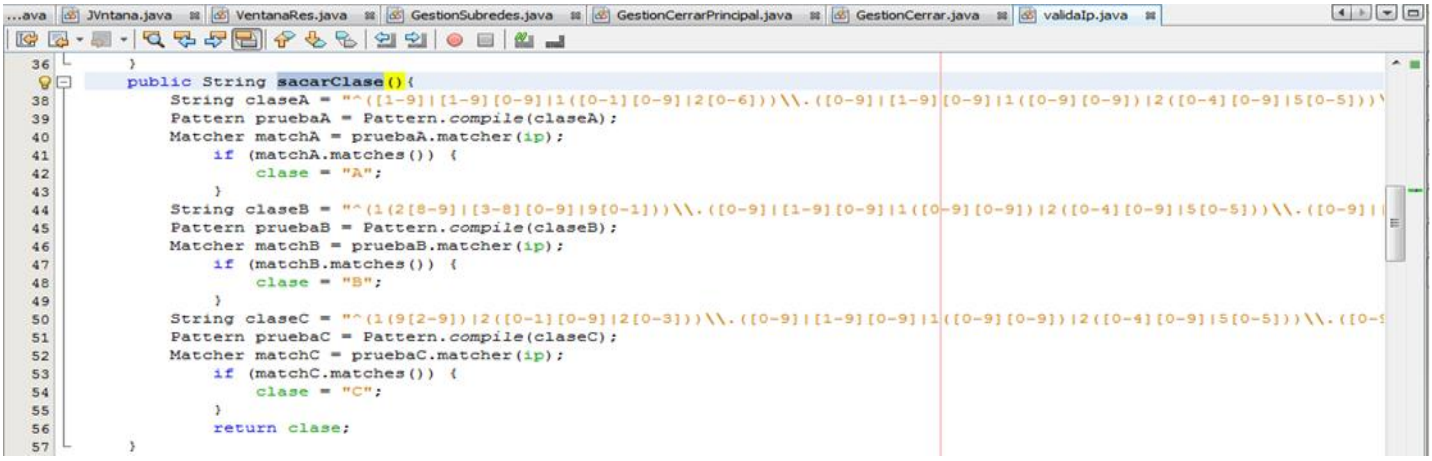
Aquí se utiliza el método que esta instanciado dentro de "ItemListener" llamado "getSelectedItem" para recoger la máscara de subred adecuada para el usuario.

- **validaIp.java**

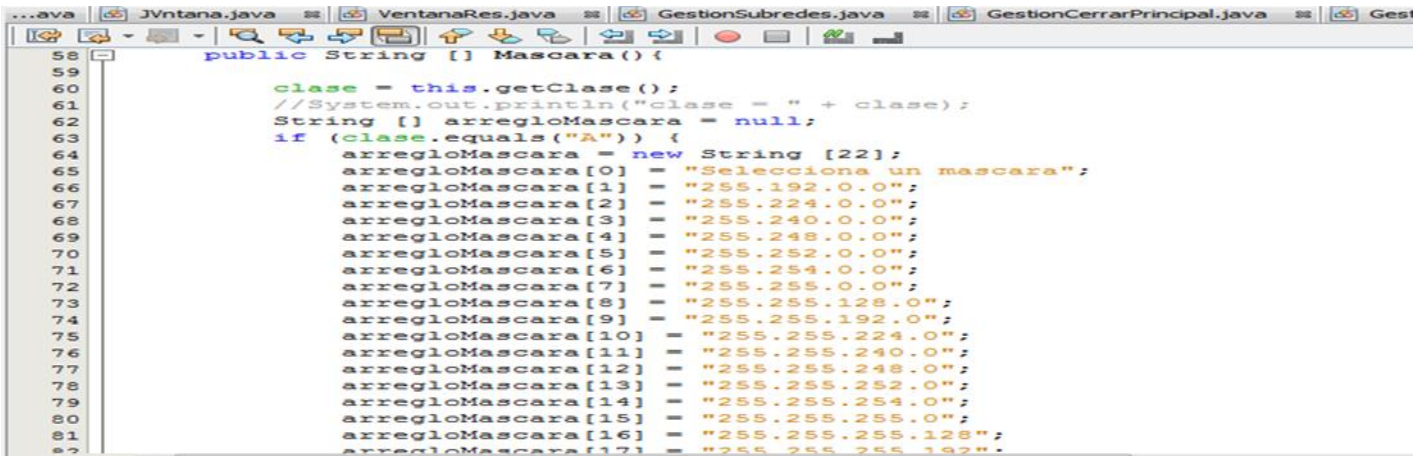
Esta clase recibe como parámetro la ip capturada del apantalla principal, la cual entra en un método de validación, el cual utiliza una expresión regular para analizar la longitud y que no sean letras.



En la clase se encuentra u método de abstracción de la clase a la que pertenece la ip recibida, llamada "sacarClase()", mediante expresiones regulares se obtiene el rango al que pertenece dependiendo del tipo de clase.



En a clase también implemento un método para la elección de las máscaras, posibles dependiendo de la clase de la ip recibida, ésta llena un arreglo con todas las posibles mascaras de subred para posteriormente mostrarlas en el combo-box que se encuentra en la ventana principal.



- **ipMascara.java**

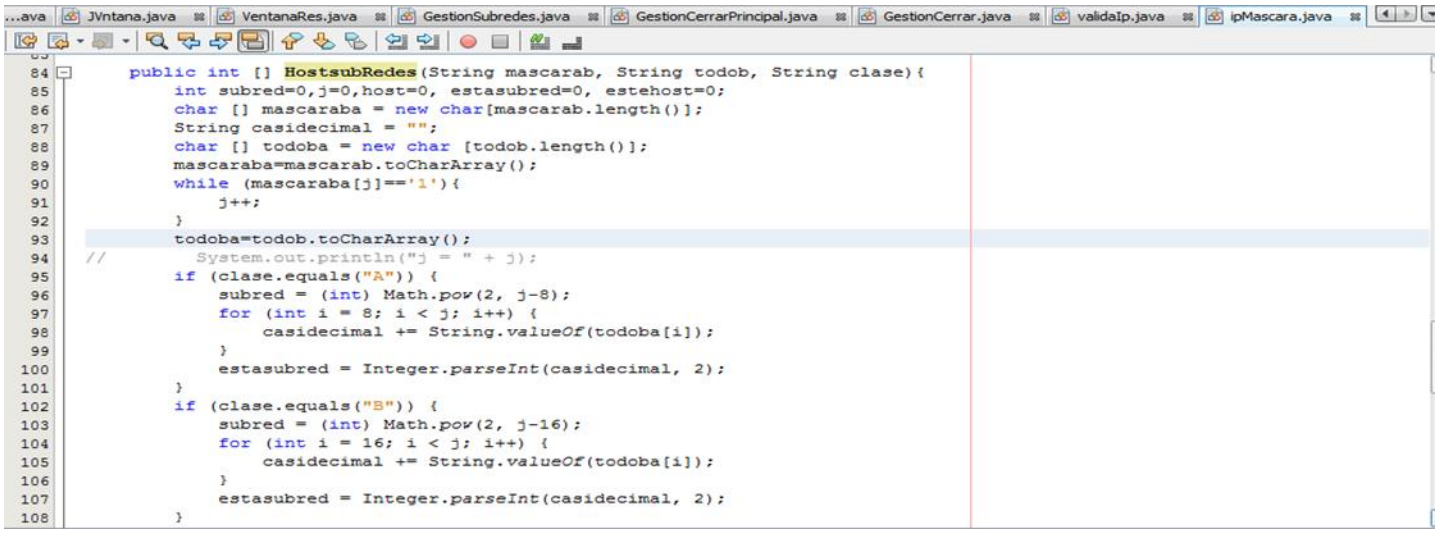
En esta clase se hacen todos los cálculos necesarios para obtener el numero de host y subredes, así como la subred y host en la cual se encuentra la ip con respecto a la máscara recibida.

Esta clase cuenta con método de:

- Conversión a binario tobinario().
- And lógica andLogica().
- Obtención de host y subredes HostsubRedes().

El método de “HostsubRedes()” es el encargado de obtener la información concerniente a los host y subredes mediante cálculos de manera binaria como se vio en la explicación teórica. Recibiendo como parámetro:

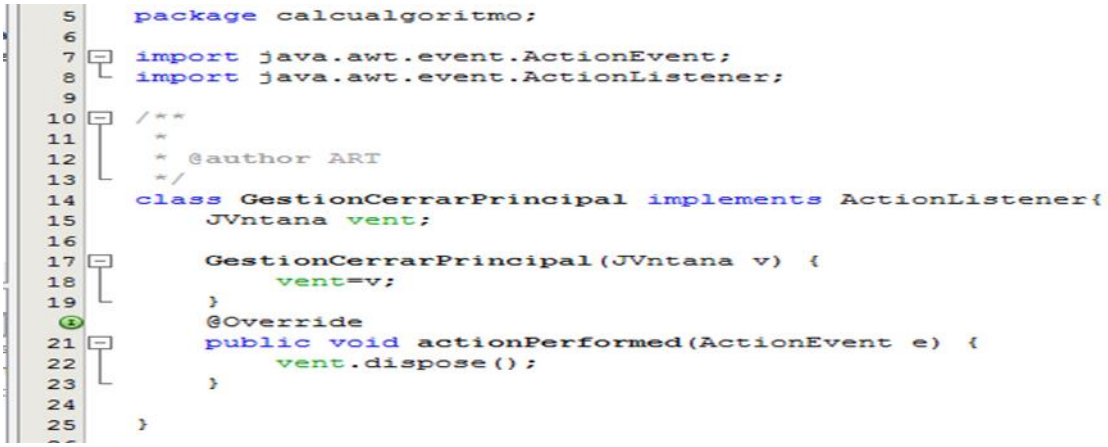
- La máscara en binario.
- La and lógica entre la máscara y la ip en binario.
- La clase de la ip.



```
84      public int [] HostsubRedes(String mascarab, String todob, String clase){
85          int subred=0,j=0,host=0, estasubred=0, estehost=0;
86          char [] mascaraba = new char[mascarab.length()];
87          String casidecimal = "";
88          char [] todoba = new char [todob.length()];
89          mascaraba=mascarab.toCharArray();
90          while (mascaraba[j]=='1'){
91              j++;
92          }
93          todoba=todob.toCharArray();
94  //          System.out.println("j = " + j);
95          if (clase.equals("A")) {
96              subred = (int) Math.pow(2, j-8);
97              for (int i = 8; i < j; i++) {
98                  casidecimal += String.valueOf(todoba[i]);
99              }
100             estasubred = Integer.parseInt(casidecimal, 2);
101         }
102         if (clase.equals("B")) {
103             subred = (int) Math.pow(2, j-16);
104             for (int i = 16; i < j; i++) {
105                 casidecimal += String.valueOf(todoba[i]);
106             }
107             estasubred = Integer.parseInt(casidecimal, 2);
108         }
```

- **GestionCerraPrincipal.java**

Esta clase solo captura el evento al dar clic en el botón cerrar de la pantalla principal, el cual cierra la ventana.



```
5   package calcualgoritmo;
6
7   import java.awt.event.ActionEvent;
8   import java.awt.event.ActionListener;
9
10  /**
11   *
12   * @author ART
13   */
14  class GestionCerrarPrincipal implements ActionListener{
15      JVntana vent;
16
17      GestionCerrarPrincipal(JVntana v) {
18          vent=v;
19      }
20      @Override
21      public void actionPerformed(ActionEvent e) {
22          vent.dispose();
23      }
24
25  }
```

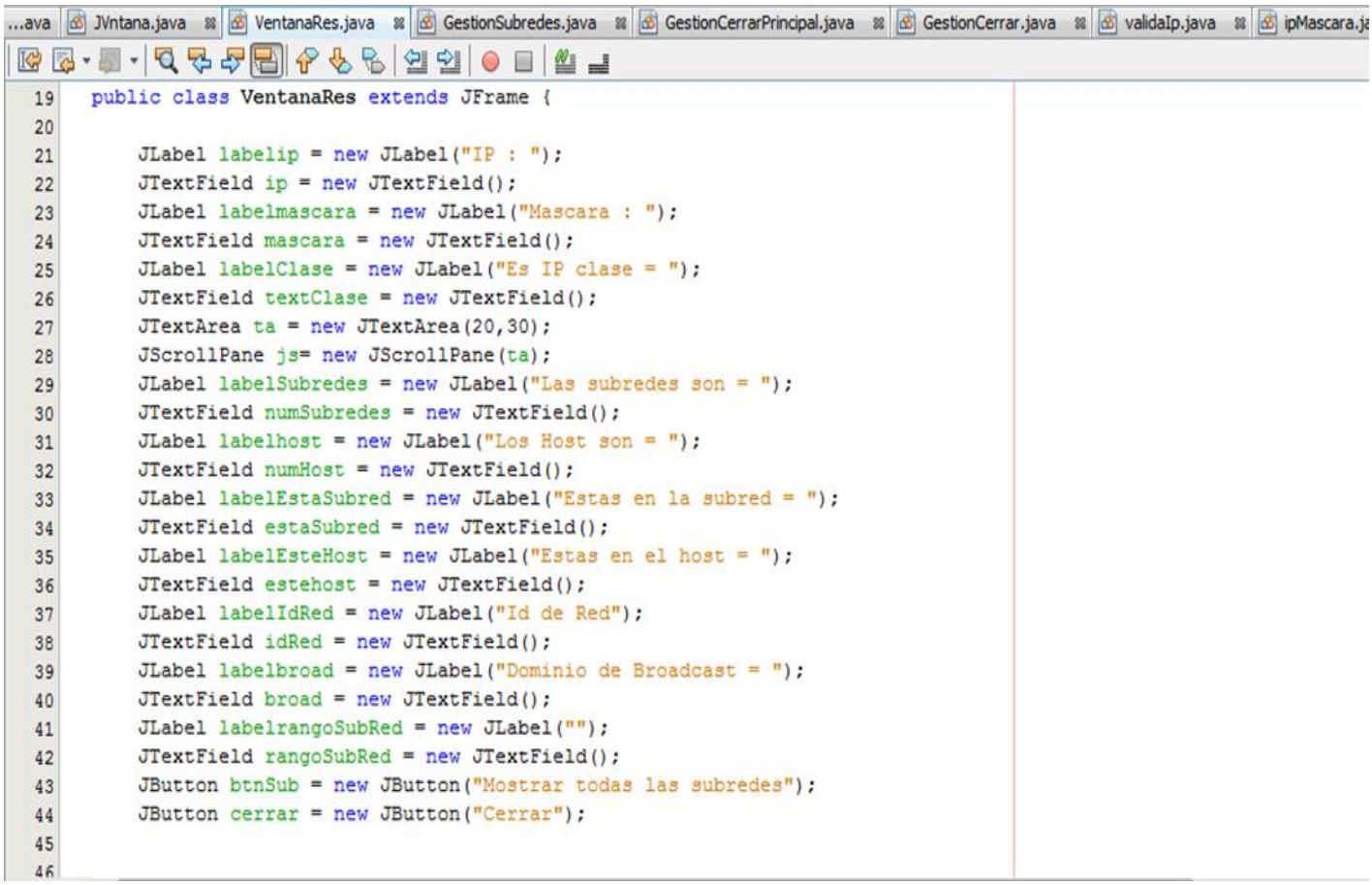
- **VentanaRes.java**

En esta clase se crea una ventana con diversos jtexfield y textLabel para poder mostrar toda la información obtenida del cálculo de subredes. La información es:

- Ip.
- Mascara.

- Clase de la ip.
- # de subredes.
- # de host en la subred.
- Subred en la que se encuentra la ip.
- # de host de la ip.
- El id de la red.
- El dominio de broadcast.
- El rango de asignación de ip en la red.

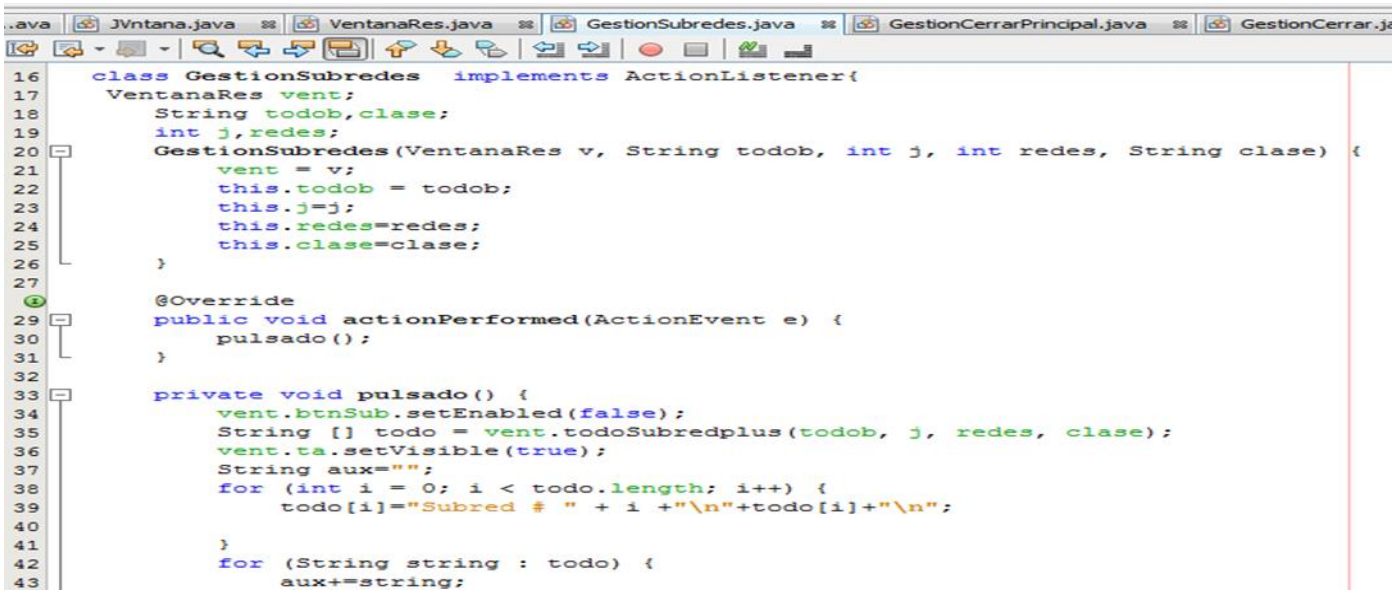
La ventana contiene un botón que muestra todas la sub-redes posibles dentro de la red, esta información es mostrada en un Jtexarea.




```
19  public class VentanaRes extends JFrame {
20
21      JLabel labelip = new JLabel("IP : ");
22      JTextField ip = new JTextField();
23      JLabel labelmascara = new JLabel("Mascara : ");
24      JTextField mascara = new JTextField();
25      JLabel labelClase = new JLabel("Es IP clase = ");
26      JTextField textClase = new JTextField();
27      JTextArea ta = new JTextArea(20,30);
28      JScrollPane js= new JScrollPane(ta);
29      JLabel labelSubredes = new JLabel("Las subredes son = ");
30      JTextField numSubredes = new JTextField();
31      JLabel labelhost = new JLabel("Los Host son = ");
32      JTextField numHost = new JTextField();
33      JLabel labelEstaSubred = new JLabel("Estas en la subred = ");
34      JTextField estaSubred = new JTextField();
35      JLabel labelEsteHost = new JLabel("Estas en el host = ");
36      JTextField estehost = new JTextField();
37      JLabel labelIdRed = new JLabel("Id de Red");
38      JTextField idRed = new JTextField();
39      JLabel labelbroad = new JLabel("Dominio de Broadcast = ");
40      JTextField broad = new JTextField();
41      JLabel labelrangoSubRed = new JLabel("");
42      JTextField rangoSubRed = new JTextField();
43      JButton btnSub = new JButton("Mostrar todas las subredes");
44      JButton cerrar = new JButton("Cerrar");
45
46
```

- **GestionSubredes.java**

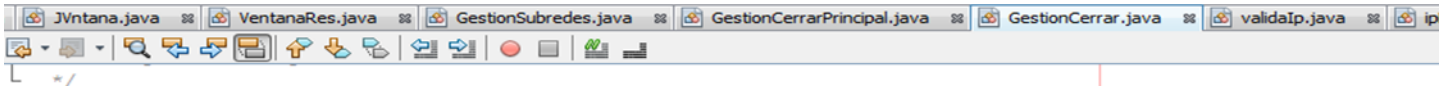
En esta clase se captura el evento del botón “Mostrar todas las subredes”, el cual tiene la misión de mostrar un “Jtextarea” con la información de todas las posibles subredes que se encuentran en la red.




```
16    class GestionSubredes  implements ActionListener{
17     VentanaRes vent;
18        String todob,clase;
19        int j,redes;
20        GestionSubredes(VentanaRes v, String todob, int j, int redes, String clase) {
21            vent = v;
22            this.todob = todob;
23            this.j=j;
24            this.redes=redes;
25            this.clase=clase;
26        }
27
          @Override
29        public void actionPerformed(ActionEvent e) {
30            pulsado();
31        }
32
33        private void pulsado() {
34            vent.btnSub.setEnabled(false);
35            String [] todo = vent.todoSubredplus(todob, j, redes, clase);
36            vent.ta.setVisible(true);
37            String aux="";
38            for (int i = 0; i < todo.length; i++) {
39                todo[i]="Subred # " + i +"\n"+todo[i]+"\n";
40
41            }
42            for (String string : todo) {
43                aux+=string;
```

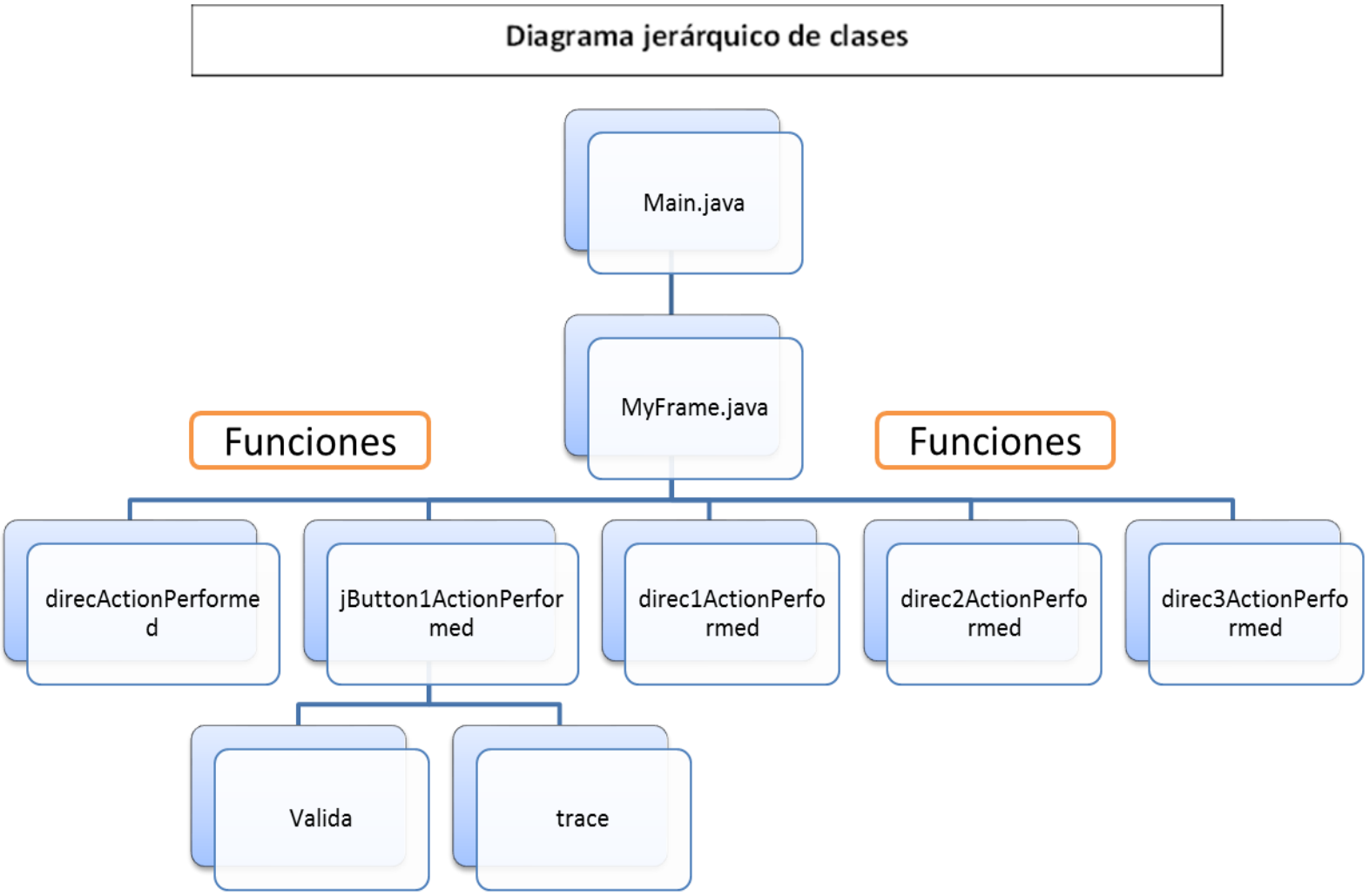
- **GstionCerrar.java**

Esta clase solamente es la encargada de cerrar la ventana de resultados para que posteriormente se abra otra venta principal de captura de datos de entrada que son la ip y la máscara de subred.




```
 */
package calcualgoritmo;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 *
 * @author ART
 */
class GestionCerrar implements ActionListener {
    VentanaRes vent;

    GestionCerrar(VentanaRes v) {
        vent=v;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        pulsado();
    }

    private void pulsado() {
        JVntana v=new JVntana("Ventana swing", 30, 80, 400, 300);
        vent.dispose();
    }

}
```

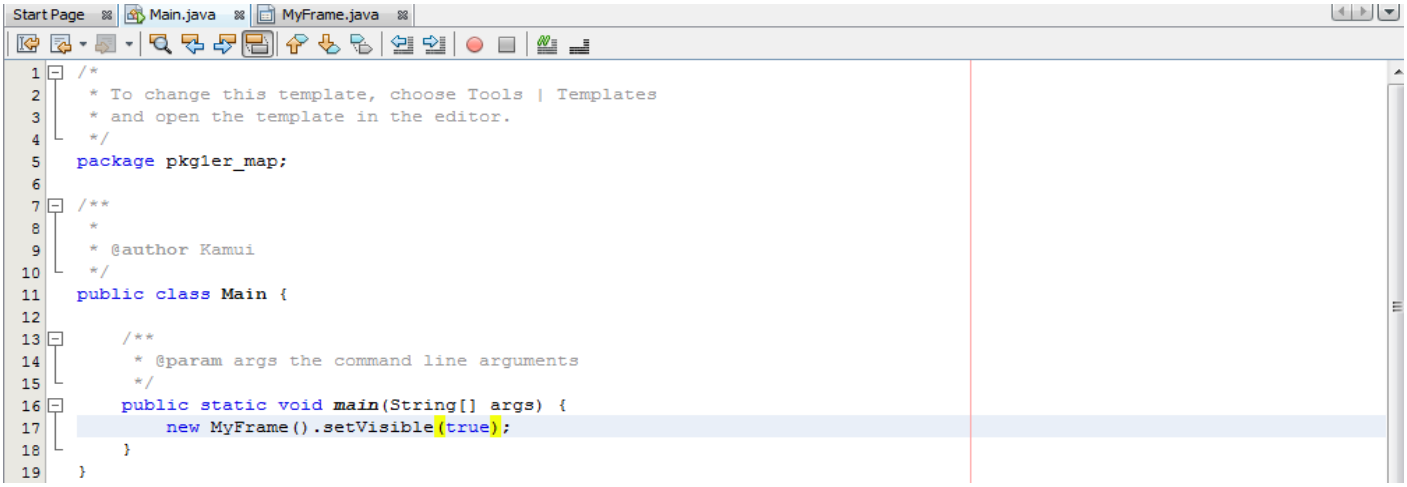
Geo-localizador de Nodos



Descripción de Clases

- **main.java**

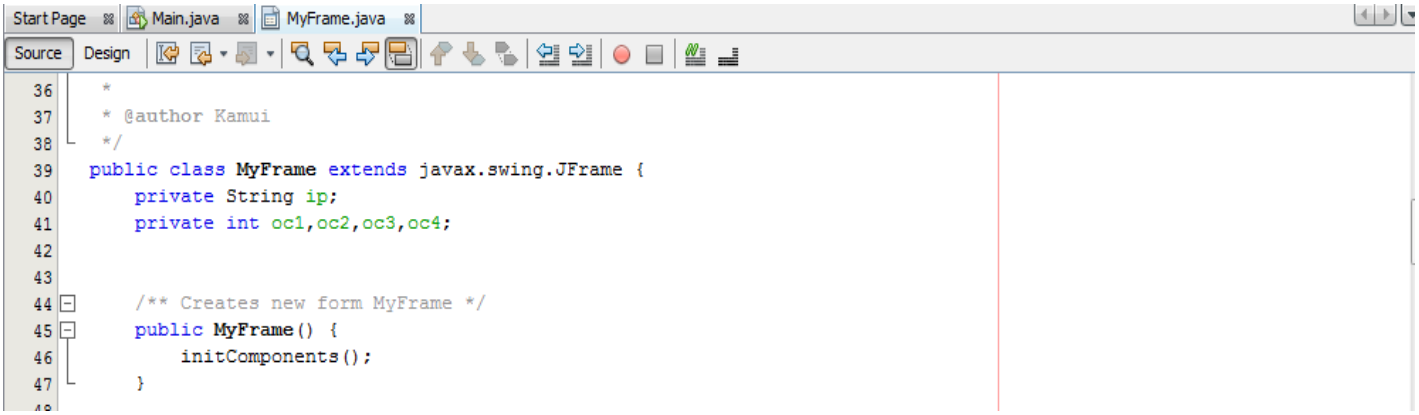
En esta clase solo se implementa la llamada a la nueva ventana que desplegará toda la información para la localización de su Host.



MyFrame.java

Función de inicialización

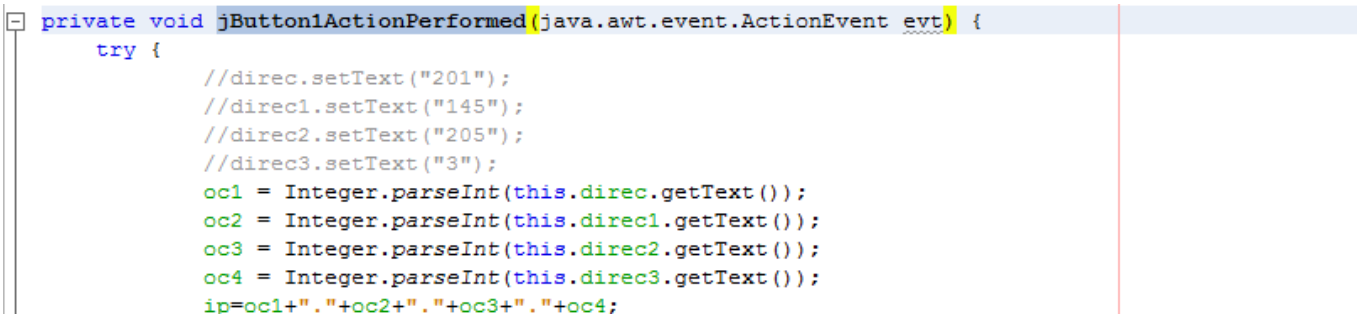
En esta clase se implementa el constructor que inicializará los valores de la ventana y la pondrán visible.



Función: jButton1ActionPerformed

En esta función se captura el evento de buscar el nodo puesto como ip en los campos de texto.

Estos obtenidos uno por uno para aplicar la función que obtiene el IPN para la relación IP-IPN-País



Función: valida

En esta función se analiza el formato de la ip que se introdujo en los campos de texto para poder verificar si está bien escrita y además contempla el rango de la ip de México, para lo cual fue establecido.

Para validar el formato de la ip se utilizaron expresiones regulares, siendo estas más eficientes, en vez de validar individualmente.

Para validar el país se cotejo el IPN con la base de datos de rangos que comprenden al país México.

Esta función recibe como parámetro:

La dirección ip completa y cada uno de los octetos para calcular el IPN.

```
315  public boolean valida(String ip, int oct1,int oct2, int oct3, int oct4
316  ////////////////////////////////////////////////////////////////////////
317      String inicio [] = new String [192];
318  inicio[ 0       ]       ="540750176";
319  inicio[ 1       ]       ="1064204288";
320  inicio[ 2       ]       ="1073036544";
321  inicio[ 3       ]       ="1103930880";
322  inicio[ 4       ]       ="1113862248";
```

La expresión regular o patrón para validación de IP:

```
String patronIp = "^([1-9]|[1-9][0-9]|1([0-9][0-9])|2([0-1][0-9]|2[0-3]))\\.([0-9]|[1-9][0-9]|1([
    Pattern prueba = Pattern.compile(patronIp);
    Matcher match = prueba.matcher(ip);
        if (match.matches() && zero==1) {
            //System.out.println("ip = " +match.group());
            return true;
        }
        else{
        return false;
        }
```

Función: trace

Esta función realiza la misión de recuperar el ISP del nodo buscado ara poderlo presentar en la pantalla. Así como lo lejos que se encuentra según los pasos dados en la traza.

Esta función recibe la ip que va a ser trazada por el comando tracert.

```
735   public String trace(String ip) throws IOException{
736   String isp = "";
737   ////////////////////////////////////////////////
738               String comando[]= new String [4];
739               comando[0]="cmd";
740               comando[1]="/c";
741               comando[2]="tracert";
742               //String ip = "173.194.64.106";
743               comando[3]=ip;
744   //            for (String string : comando) {
745   //                System.out.println("string = " + string);
746   //         }
```

Estructura de la base de datos

En este paso la base de datos es una relación entre una ip y una designación a un país.

Esta base de datos contiene prácticamente todos los países, aunque haya una actualización muy a menudo.

Implementación

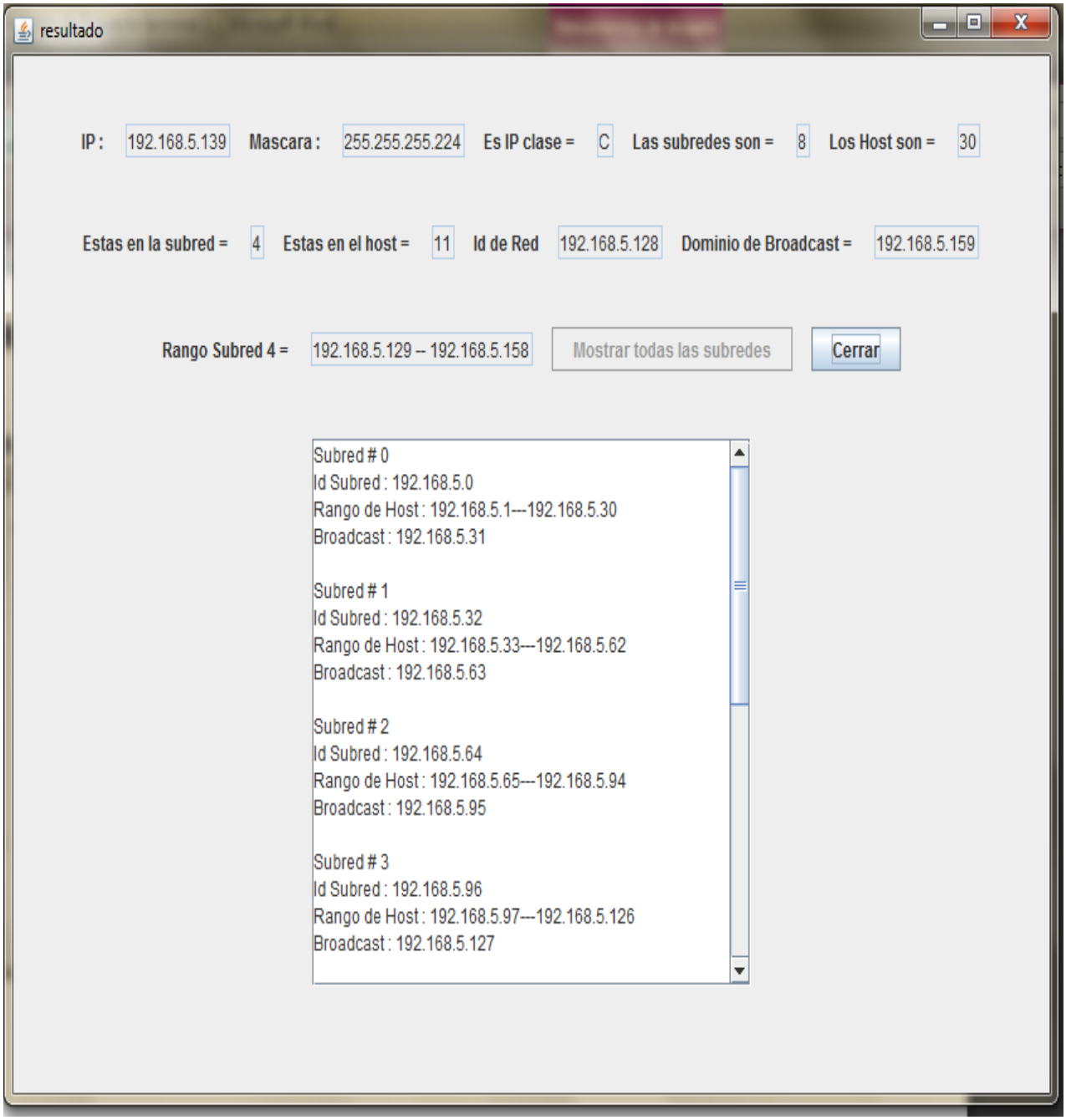
Para la implementación utilizó el IDE NetBeans además de Wireshark (Ethereal).

5. Resultados y alcances del proyecto

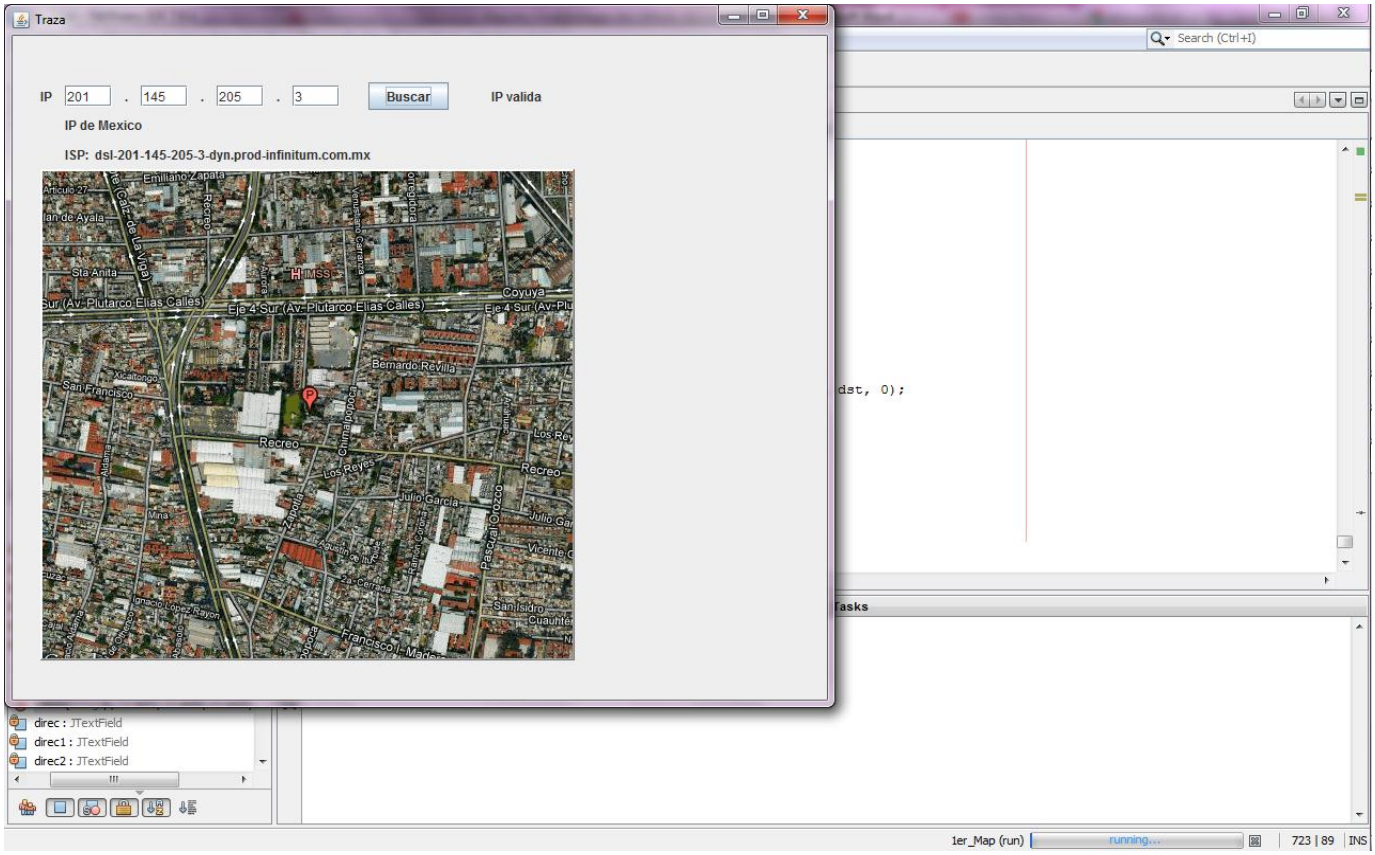
Resultados

Los resultados fueron los esperados en lo que respecta a las pruebas d escritorio y enseguida unos pantallazos:

Calculadora de Subredes



Geo-Localizador de Host.



Con el software libre WireShark observamos la comunicación cuando realizamos el tracert:

Network Protocol Analyzer

Version 1.6.1 (SVN Rev 38096 from /trunk-1.6)

Copyright 1998-2011 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

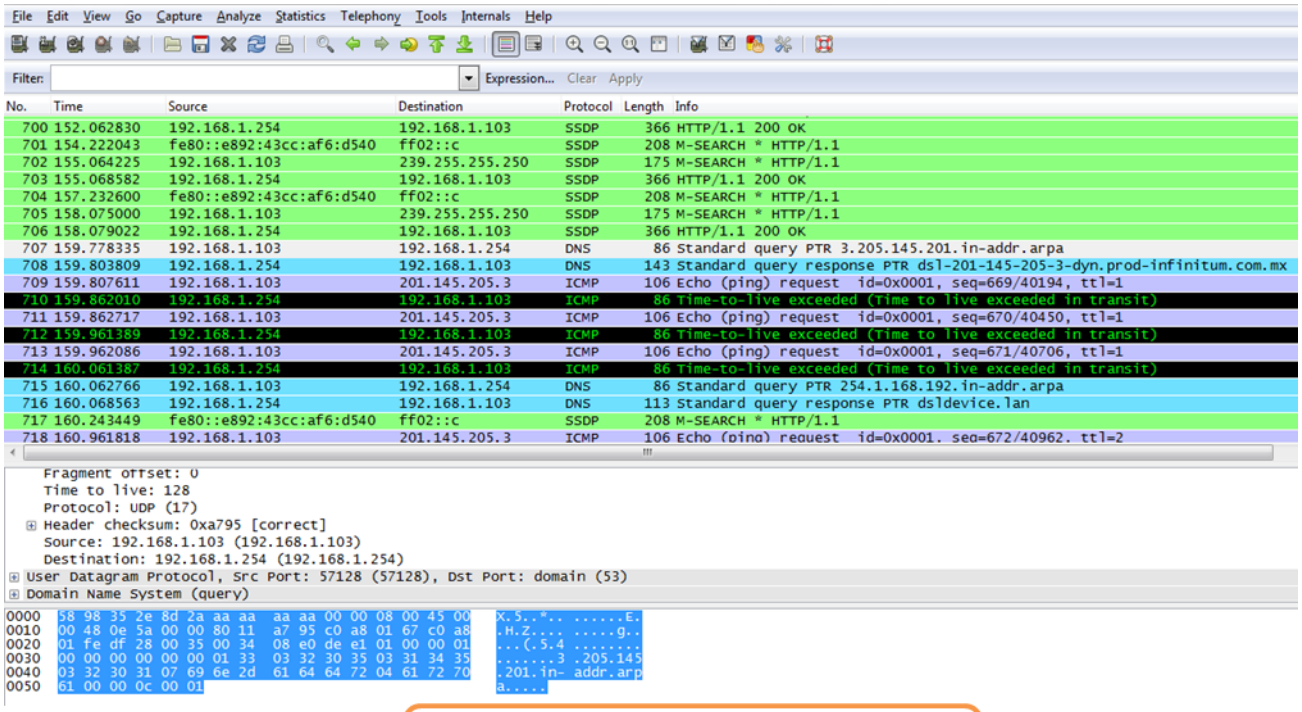
Compiled (32-bit) with GTK+ 2.22.1, with GLib 2.26.1, with WinPcap (version unknown), with libz 1.2.5, without POSIX capabilities, without libpcre, with SMI 0.4.8, with c-ares 1.7.1, with Lua 5.1, without Python, with GnuTLS 2.10.3, with Gcrypt 1.4.6, with MIT Kerberos, with GeoIP, with PortAudio V19-devel (built Jul 18 2011), with AirPcap.

Running on 32-bit Windows 7, build 7600, with WinPcap version 4.1.2 (packet.dll version 4.1.0.2001), based on libpcap version 1.0 branch 1_0_rel0b (20091008), GnuTLS 2.10.3, Gcrypt 1.4.6, without AirPcap.

Built using Microsoft Visual C++ 9.0 build 21022

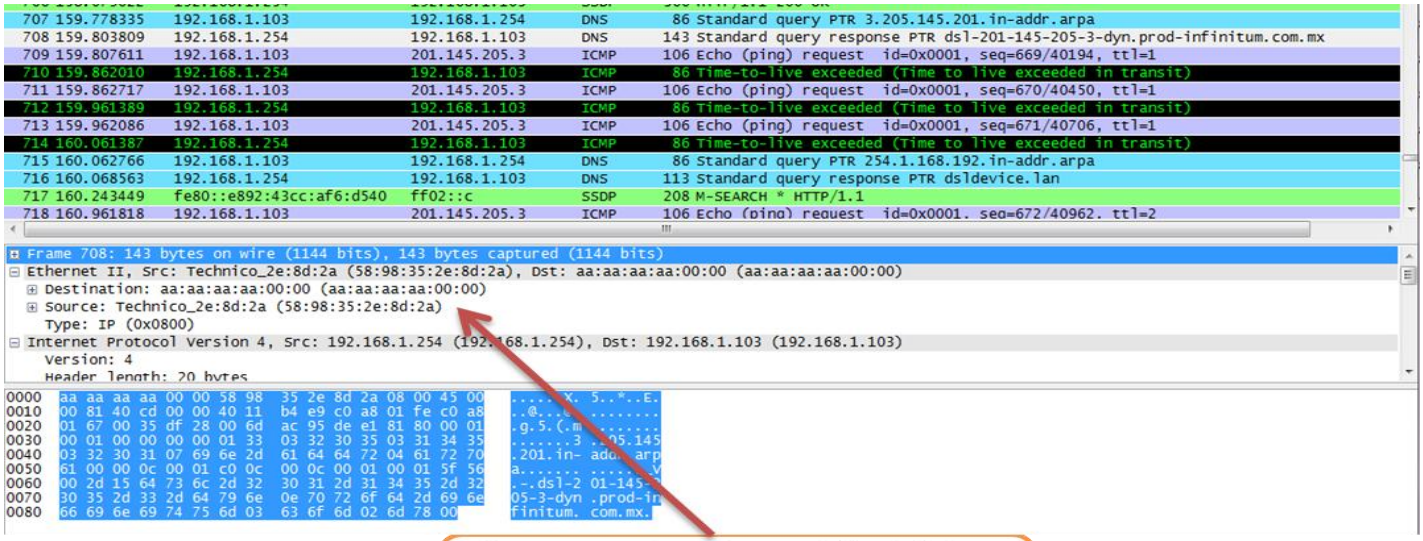
Wireshark is Open Source Software released under the GNU General Public License.

Check the man page and http://www.wireshark.org for more information.



Captura 1: Response DNS

En esta pantalla notamos la petición del ping del ordenador local y del remoto como un paquete ICMP todos con un TTL=1.
También se observa el ISP del nodo buscado.



Captura 2: Dirección Física (MAC Address)

Aquí se observa la dirección física del nodo buscado y la propia (aa:aa:aa:aa:00:00).

6. Conclusiones y perspectivas del proyecto

Concluyendo, Fue un trabajo que rindió frutos y que demuestra los conocimientos que he adquirido en esta institución, y es de varias áreas de concentración por cierto, espero y sea de alguna ayuda didáctica o profesional para generaciones posteriores.

Aún falta expandir el horizonte de los países, para próximas versiones, así como que sea concurrente en el sentido de la demanda, pero sé que próximamente se verá reflejado este proyecto en algunas ideas que están por salir a flote y así poder conjuntar dichas ideas.

7. Bibliografía

[1]. M. Velázquez, "Análisis y Estudio de los protocolos de ruteo interno en plataforma Cisco", Proyecto terminal de Ingeniería Electrónica, Universidad Autónoma Metropolitana Azcapotzalco, D.F, México, 2007

[2]. adeona.cs.washington.edu/ [En línea]. Disponible: http://adeona.cs.washington.edu/

[3]. www.ip-adress.com [En línea]. Disponible: www.ip-adress.com/ip_tracer/

[4]. www.internic.net/ [En línea]. Disponible: *www.internic.net/*

[5]. R. Malhotra, *IP routing,* 1a ed., C.A., O'Reilly & Associates Inc., 2002.

[6]. B. Shneiderman y C. Plaisant, *Designing the user interface,* 5a ed., Maryland, Addison Wesley & Pearson, 2010.

[7]. F. Ceballos, *Java 2: Interfaces gráficas y aplicaciones para Internet*, 3a ed., D.F., México, Alfaomega, 2008.