



LOCALITZADOR GRÀFIC D'ADRECES IP

“GEOLOCALITZACIO IP”

Guillermo Luque Arenas
Grau d'Enginyeria Informàtica de Sistemes

Maria Isabel March Hermo

Juny 2018



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Localitzador Gràfic d'Adreces IP. "Geolocalització IP"</i>
Nom de l'autor:	<i>Guillermo Luque Arenas</i>
Nom del consultor:	<i>Maria Isabel March Hermo</i>
Data de lliurament (mm/aaaa):	<i>06/2018</i>
Àrea del Treball Final:	<i>Xarxes de Computadors</i>
Titulació:	<i>Grau d'Enginyeria Informàtica de Sistemes</i>
Resum del Treball:	
<p>Aquesta memòria recull el procés d'elaboració d'un producte destinat a la geolocalització d'equips a través de la seva adreça IP. Per tal de poder assolir l'objectiu, s'ha dividit aquest procés en quatre blocs.</p> <p>El primer, es centra en recollir i fer una explicació de tots aquells elements que regulen i conformen la xarxa, com els protocols de comunicació o com es distribueix l'adreçament IP de la xarxa.</p> <p>El segon, desenvolupa un anàlisi de quins són els recursos que es poden utilitzar per desenvolupar el producte. En concret, s'analitzen diferents serveis de geolocalització IP que es poden trobar a la xarxa, també s'avaluen alguns dels llenguatges de programació que poden ser utilitzats per acomplir l'objectiu de l'aplicació, igualment es fa un anàlisi d'alguns serveis de representació de mapes que hi ha a Internet, com són <i>Google Maps</i> i <i>Open Street Maps</i>. Finalment, es realitza una explicació de les opcions triades per a implementar l'aplicació.</p> <p>En el tercer bloc, s'exposa com s'ha dissenyat el producte tenint en compte els diferents aspectes que ha de cobrir una aplicació. També, s'inclou la part de disseny més concreta, com és el diagrama de classes que mostra com es vol implementar l'aplicació.</p> <p>Finalment, el quart bloc, fa una explicació de com s'ha portat a la pràctica el que s'ha dissenyat. Aquesta es divideix en el procés per trobar els nodes intermedis, o traça, la consulta de la informació de cada node, i com s'ha representat a través d'un servei de mapa la ruta i els nodes trobats.</p>	

Abstract:

This report includes the process of preparing a product for the geolocation of equipment through its IP address. In order to achieve the goal, this process has been divided into four blocks.

The first one focuses on collecting and explaining all those elements that regulate and shape the network, such as communication protocols or how to distribute IP address of the network.

The second one develops an analysis of the resources that can be used to develop the product. In particular, we analyze different IP geolocation services that can be found on the network, some of the programming languages that can be used to accomplish the purpose of the application are also evaluated, some analyzes are also made Map representation services on the Internet, such as Google Maps and Open Street Maps. Finally, an explanation of the options chosen to implement the application is made.

In the third block, we explain how the product was designed taking into account the different aspects that an application must cover. Also, the more specific design part is included, such as the class diagram that shows how the application is to be implemented.

Finally, the fourth block gives an explanation of how the design has been put into practice. This is divided into the process to find the intermediate nodes, or trace, the query for the information of each node, and how the route and nodes found have been represented through a map service.

Paraules clau:

adreçament, IP, geolocalització, DNS, *Java*, *Google*, *ping*, *traceroute*

Índex

1.	Introducció	1
1.1.	Justificació del TFG i context en el qual es desenvolupa: punt de partida i aportació del TFG	2
1.2.	Objectius del TFG	2
1.3.	Enfocament i mètode seguit	3
1.4.	Planificació del projecte	4
1.5.	Productes obtinguts	7
1.6.	Breu descripció dels altres capítols de la memòria.....	7
2.	Marc teòric	9
2.1.	Encapsulament i la pila TCP/IP	9
2.2.	Adreçament IPv4	10
2.3.	Protocol ICMP (Protocol de Missatges de Control d'Internet)	12
2.4.	Ping.....	14
2.5.	Traceroute	15
2.6.	Domain Name System (DNS).....	15
2.7.	Whois.....	15
3.	Anàlisi de recursos.....	17
3.1.	Geolocalització IP	17
3.2.	Llenguatges de programació	21
3.3.	Serveis de Mapes	23
3.4.	Elecció de recursos.....	26
4.	Disseny de l'aplicació	28
4.1.	Cas d'ús.....	28
4.2.	Diagrama d'Estats.....	28
4.3.	Diagrama de seqüència	30
4.4.	Diagrama de classes	31
4.5.	Disseny de les finestres	33
5.	Implementació del producte.....	35
5.1.	Creació de paquets i traça.....	35
5.2.	Geolocalització IP dels nodes.	37
5.3.	Mostrar informació en un mapa	38
5.4.	Informació <i>Whois</i>	39

5.5. Entorn gràfic de l'aplicació	40
6. Conclusions.....	43
7. Glossari.....	44
8. Bibliografia	45
9. Annexos.....	46
9.1. Annex A. Manual de instal·lació.	46
9.2. Annex B. Manual d'usuari.	52

Índex d'Il·lustracions

Il·lustració 1. Diagrama de Gantt.....	7
Il·lustració 2. Pila TCP/IP.....	10
Il·lustració 3. Estàndard OSI.....	10
Il·lustració 4. Estructura datagrama IP	12
Il·lustració 5. Capçalera datagrama IP del protocol ICMP	13
Il·lustració 6. Tarifes MaxMind.....	18
Il·lustració 7. Tarifes IP2Location.....	18
Il·lustració 8. Tarifes Ipinfo.io	19
Il·lustració 9. Pantalla de selecció de camps	21
Il·lustració 10. Entorn PyCharm.....	22
Il·lustració 11. Entorn IDE Netbeans 8.2.....	23
Il·lustració 12. Open Street Map	24
Il·lustració 13. Google Static Maps API.....	25
Il·lustració 14. Google Maps JavaScript API.....	25
Il·lustració 15. Tarifes Google Maps JavaScript	26
Il·lustració 16. Cas d'us	28
Il·lustració 17. Diagrama d'estats	29
Il·lustració 18. Diagrama de seqüència	30
Il·lustració 19. Diagrama de classes.....	31
Il·lustració 20. Disseny finestra principal.....	33
Il·lustració 21. Finestra Whois	39
Il·lustració 22. Finestra principal (Inici)	40
Il·lustració 24. Finestra principal durant el procés de traça.....	41
Il·lustració 25. Pantalla principal al finalitzar el procés de traça.....	41
Il·lustració 26. WinPcap, inici d'instal·lació.	47

Il·lustració 27. WinPcap, acceptació de llicència.....	47
Il·lustració 28. WinPcap, configuració.....	48
Il·lustració 29. WinPcap, finalització d'instal·lació.	48
Il·lustració 30. Jpcap, inici instal·lació.....	49
Il·lustració 31. Jpcap, benvinguda instal·lador.	49
Il·lustració 32. Jpcap, acceptació de llicència.	50
Il·lustració 33. Jpcap, procés d'instal·lació.	50
Il·lustració 34. Jpcap, finalització del procés.	51

Índex de Taules

Taula 1. Dates d'entrega.....	4
Taula 2. Classes IP.....	12
Taula 3. Rangs d'adreces privades.....	12
Taula 4. Respostes ICMP	14
Taula 5. Servidors Whois dels diferents continents	16
Taula 6. Taula de camps de IP-Api.....	20
Taula 7. Missatges d'error de Ip-api	20

1. Introducció

Els inicis d'Internet van ser a la dècada dels anys 1960, als Estats Units, quan l'organització ARPA (*Advanced Research Project Research*) va crear una xarxa per tal d'aprofitar millor el potencial dels ordinadors i alhora permetre que els laboratoris i investigadors poguessin disposar dels seus propis equips. Per tal d'evitar duplicitats en la feina i pèrdua de recursos es va crear la xarxa *ARPAnet*, la qual va permetre a aquests, estar interconnectats i aprofitar millor els recursos. Poc a poc, es van anar afegint a aquesta xarxa més organitzacions governamentals i universitats, aquesta xarxa ofería una interconnexió d'alta velocitat entre tots els participants i donava accés a la informació d'altres organitzacions d'una manera molt eficient.

Des del seu inici oficial, al any 1969, quan es va fer la interconnexió entre les universitats de UCLA i Stanford mitjançant una xarxa commutada telefònica, fins als anys 1990, es pot considerar que *ARPAnet* va estar en constant construcció, però és a partir dels anys 1990 que amb la introducció de noves facilitats de connexió i eines gràfiques simples per facilitar l'ús de la xarxa, aquesta es popularitza. També en el any 1990 es crea el primer client *Web* anomenat *WorldWideWeb* (*www*) i comença a anomenar-se Internet, tal com es coneix actualment.

Actualment, a Internet es calcula que hi ha 4.021 mil milions d'usuaris, la xarxa s'ha globalitzat i aquest número representa el 53% de la població mundial. La xarxa s'ha convertit per a molta gent i negocis en un punt d'interconnexió entre aquests, i cada vegada han anat sorgint noves necessitats a l'hora de millorar aquesta relació.

Una d'aquestes necessitats és poder saber des de quin punt geogràfic accedeix un usuari o organització. La xarxa ha proporcionat als usuaris la possibilitat d'accedir, en principi, a qualsevol recurs, independentment d'on es troba aquest, però pot ser que aquest recurs no hagi de ser accessible per a segons quines regions geogràfiques. Per exemple, pot existir una tenda virtual que no tingui distribució per a Àsia, o que per a segons quins països ofereixi uns productes diferents, llavors apareix la necessitat de saber des de quin punt geogràfic està accedint aquest usuari per tal d'optimitzar els recursos d'aquest determinat servei. Altra possibilitat pot ser per motius de seguretat.

Es en aquest punt apareix la necessitat de la Geolocalització IP, aquesta es tracta bàsicament, de relacionar una adreça IP amb un punt concret de la geografia mundial. Per cobrir aquesta necessitat s'han desenvolupat diversos serveis dedicats a facilitar amb major o menor exactitud les dades de geolocalització de les adreces IP.

1.1. Justificació del TFG i context en el qual es desenvolupa: punt de partida i aportació del TFG

Una de les parts que sempre m'ha interessat dels entorns informàtics ha sigut com s'interconnecten els equips i comparteixen la informació entre ells. Internet ha aportat als equips informàtics la possibilitat de compartir informació eliminant moltes fronteres geogràfiques, és per aquest motiu que vaig triar aquest camp per desenvolupar el Treball de Fi de Grau.

Durant l'estudi de les diferents assignatures de Xarxes de Computadors he pogut aprendre moltes de les característiques d'aquestes comunicacions entre equips, des del punt concret d'un equip fins al funcionament d'una xarxa complexa com pot ser una Intranet o Internet. També he après el funcionament dels diferents protocols de comunicació i els elements importants que formen una xarxa, com són els encaminadors, els commutadors o els servidors de domini (DNS).

És a partir d'aquest estudi que apareix la possibilitat d'ampliar els coneixements de com s'organitza concretament Internet i com es poden aprofitar els diferents serveis que existeixen a la xarxa per a desenvolupar una aplicació que permeti geolocalitzar els equips dintre del món físic, és a dir, en un mapa geogràfic.

Aquest Treball de Fi de Grau parteix dels coneixements previs sobre xarxes, i pretén ampliar aquests coneixements aportant una visió de les possibilitats que ofereix la xarxa per a utilitzar recursos i combinar-los per a elaborar un producte que a la vegada ofereixi un servei concret, en aquest cas la geolocalització d'equips a través de la seva adreça IP. L'aportació que pretén és la clarificació de conceptes de geolocalització i protocols de comunicació que permeten la identificació dels nodes existents en una comunicació en una xarxa.

1.2. Objectius del TFG

L'objectiu d'aquest TFG és crear un producte que, donat un nom de host d'Internet proporcioni la informació corresponent a aquest, a més de mostrar per quins nodes han de passar els paquets per tal d'accedir-hi. La informació relativa als nodes serà la seva geolocalització dins d'un mapa mundi i aquests aniran enllaçats amb una línia per a mostrar el camí de la informació.

Per tal de portar a terme aquestes accions, l'aplicació ha de realitzar diferents processos enllaçats entre si. Primer, s'ha d'implementar un mètode que retorni la informació dels nodes a mode de *traceroute*, tot creant els paquets que utilitza el protocol ICMP per a realitzar aquest. A continuació, s'hauran de situar aquests nodes en unes coordenades geogràfiques per tal de, finalment, poder situar-los dintre d'un mapa mundi. A banda d'aquest procés de geolocalització, es vol fer una consulta al servei *Whois* per tal que retorni tota la informació relativa al nom de host indicat si l'usuari ho demana.

Els objectius generals són els següents:

- Crear un producte capaç d'identificar els nodes que hi ha entre l'equip des d'on s'està executant i un nom de host o adreça IP d'Internet.
- Implementar dintre del producte la possibilitat de consultar les dades referents al host de destí.
- Ubicar dintre d'un mapa mundi els nodes identificats amb anterioritat com a nodes entre l'equip origen i el host de destí.
- Realitzar la memòria que reculli la documentació elaborada durant tot el procés del projecte, així com l'explicació del funcionament d'aquest.
- Aprofundir a nivell teòric i pràctic en els conceptes relacionats amb el projecte, com poden ser l'estructura dels datagrames ICMP i les seves utilitats, les diferents possibilitats que existeixen per tal de geolocalitzar un host determinat, o la implementació d'aquests en el llenguatge de programació triat.

1.3. Enfocament i mètode seguit

A l'hora de planificar la construcció d'aquesta aplicació que ha de mostrar la ruta que hi ha entre dos equips, un d'origen i un altre de destí, s'ha optat per dividir el conjunt de funcionalitats que s'han de implementar i treballar cadascuna d'elles per separat, però alhora d'una manera incremental. És a dir, s'ha de resoldre cada funcionalitat per separat i una vegada es té implementada, aquesta ens proporcionava la base de la següent.

Les funcionalitats necessàries per completar el projecte són les següents:

- Realització de la traça entre el node origen i el destí.
- Obtenció de tota la informació necessària dels nodes, com és la latitud i la longitud.
- Mostrar aquests nodes en una taula.
- Visualització d'aquests nodes, juntament amb la ruta que realitzen els paquets, en un mapa.

Així doncs, primerament caldrà construir el mètode que permeti realitzar la traça entre nodes per tal d'obtenir la informació bàsica d'aquests.

Una vegada es tingui els nodes identificats amb la informació obtinguda en el pas anterior, caldrà consultar en algun servei la informació concreta de geolocalització de cada node i afegir-la a la informació que ja es té.

A continuació, cal mostrar la informació obtinguda de cada node en una taula que permeti a l'usuari poder revisar aquesta informació d'una manera fàcil i intuïtiva. Això es farà mitjançant una taula que recollirà totes les dades obtingudes.

Finalment, utilitzant les dades obtingudes en el pas anterior, caldrà utilitzar un servei de mapes per a mostrar-la correctament en un mapa, de tal manera que també quedi representada la ruta entre nodes.

1.4. Planificació del projecte

1.4.1. Dates d'entrega

Data	Lliurament
15 d'abril	PAC 1
27 de maig	PAC 2
10 de juny	Memòria del Producte TFG i Producte finalitzat
17 de juny	Presentació del TFG
18-24 de juny	Preguntes del tribunal

Taula 1. Dates d'entrega

1.4.2. Fites i temporització

Tasca 1

Temporització: 1 setmana (5 de març al 11 de març)

Descripció: Recollida i classificació de tota la informació per a generar el marc teòric en el que es basa el TFG.

Tasques:

- Recollida d'informació relativa als protocols necessaris per a realitzar les consultes amb els diferents nodes (*ICMP Echo Reply* i *ICMP Echo Request*).
- Elaboració del marc teòric en el qual es fonamentarà el projecte.
- Recollida d'informació sobre les consultes als serveis *Whois*.
- Anàlisi de les diferents possibilitats a l'hora d'obtenir la informació relativa a la geolocalització dels nodes.

Fites:

- Elaborar un document que reculli tota aquesta informació estructurada: descripció del protocol ICMP, diferents serveis que existeixen per a obtenir la geolocalització d'un node, informació de com s'estructura el servei *Whois*.
- Afegir la informació elaborada a la memòria.

Tasca 2

Temporització: 1 setmana (12 de març al 18 de març)

Descripció: Identificar les necessitats concretes a l'hora d'implementar el TFG i triar les eines necessàries per a satisfer aquestes.

Tasques:

- Avaluar diferents llenguatges de programació valorant les seves possibilitats a l'hora d'implementar datagrames ICMP *Echo Replay* i *Echo Request*, i les seves funcionalitats per a treballar amb objectes i finestres.
- Avaluar quin sistema d'informació sobre la geolocalització és la més adient per a implementar el TFG.
- Avaluar els diferents serveis existents per a poder visualitzar dintre d'un mapa mundi els diferents nodes la ruta que segueixen els paquets entre aquests.

Fites:

- Definir el marc pràctic que ens servirà per a desenvolupar el projecte de forma eficient.
- Establir el llenguatge de programació que s'utilitzarà i el tipus d'aplicació que es vol implementar.
- Afegir la informació elaborada a la memòria.

Tasca 3

Temporització: 1 setmana (19 de març al 25 de març)

Descripció: Disseny dels mòduls i processos que ha de tenir el producte per a realitzar les tasques desitjades.

Tasques:

- Disseny de l'estructura concreta dels mòduls necessaris per a satisfer les necessitats del producte.
- Disseny de les finestres i funcionalitats que formaran part del producte.

Fites:

- Disseny de l'aplicació amb l'anàlisi de les decisions preses.
- Afegir la informació elaborada a la memòria.

Tasca 4

Temporització: 7 setmanes (26 de març al 13 de maig)

Descripció: Implementar amb el llenguatge de programació triat l'aplicació que compleixi els objectius marcats al disseny.

Tasques:

- Implementar estructura bàsica de les finestres necessàries per a la funcionalitat de l'aplicació. (26 de març al 31 de març)
- Implementar translació del nom del host a adreça IP. (1 d'abril al 3 d'abril)
- Implementar consulta ICMP per obtenir ruta i nodes. (4 d'abril al 15 d'abril)

- Implementar consulta servei *Whois*. (16 d'abril al 22 d'abril)
- Implementar consulta de la latitud i la longitud al servei de geolocalització triat. (23 d'abril al 29 d'abril)
- Implementar representació de les dades obtingudes en el API de mapes triat. (30 d'abril al 13 de maig)

Fites:

- Aplicació finalitzada per a poder començar proves d'usuari.
- Afegir la informació elaborada a la memòria.
-

Tasca 5

Temporització: 1 setmana (14 de maig al 20 de maig)

Descripció: Realitzar proves d'usuari per tal de detectar possibles errades, documentar aplicació.

Tasques:

- Proves de les diferents funcionalitats de l'aplicació.
- Anàlisi del producte final.

Fites:

- Aplicació finalitzada.
- Afegir la informació elaborada a la memòria.

Tasca 6

Temporització: 3 setmanes (21 de maig al 10 de juny)

Descripció: Finalització de la memòria i documentació necessària per a fer l'entrega de TFG.

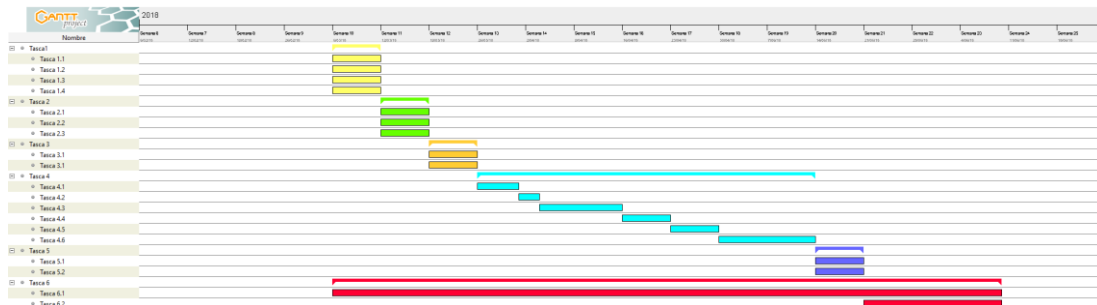
Tasques:

- Finalitzar la memòria afegint la descripció final de l'aplicació i el manual de funcionament.
- Realitzar presentació del producte.

Fites:

- Memòria del TFG.
- Presentació del TFG.

1.4.3. Diagrama de Gantt



Il·lustració 1. Diagrama de Gantt

1.5. Productes obtinguts

Al finalitzar el projecte s'hauran elaborat dos productes que formaran part de l'entrega del TFG, aquests productes seran l'aplicació *GeolocalitzacióIP* i la memòria on es recollirà tot el procés de treball per arribar a la finalització d'aquesta.

L'aplicació *GeolocalitzacióIP* és un programari que a partir d'una adreça IP o el nom d'un domini, ens proporcioni la informació relativa a la ruta que segueixen els paquets entre l'equip origen i l'equip destí. Mitjançant la informació recollida es mostrarà a través d'un mapa aquesta ruta entre nodes. Igualment tota la informació es recollirà en una taula que es podrà visualitzar en tot moment.

La memòria final, recollirà la base teòrica en la qual es fonamenta, així com totes les fases del disseny i implementació del producte.

1.6. Breu descripció dels altres capítols de la memòria

A continuació es troben els capítols de desenvolupament del projecte concret. La distribució d'aquests s'ha fet amb una introducció del marc teòric per després entrar a desenvolupar una part més pràctica fins arribar al desenvolupament del producte final. Els capítols són els següents:

- **Capítol 2, Marc Teòric**, en aquest es fa una explicació dels elements teòrics que formen part de la comunicació dintre d'una xarxa d'equips informàtics i quins són els protocols bàsics que s'utilitzaran per aconseguir els objectius del producte.
- **Capítol 3, Anàlisi de Recursos**, es fa una avaluació dels diferents recursos disponibles a la xarxa per tal d'obtenir les diferents funcionalitats del producte, així com d'alguns dels diferents llenguatges de programació que es poden utilitzar per a implementar aquest producte. Finalment es fa la tria i justificació dels recursos triats.
- **Capítol 4, Disseny de l'Aplicació**, es realitza el disseny de les funcionalitats i comportament que tindrà el producte. Aquest es realitza tenint en compte els diferents digrames que permeten concretar quin comportament ha de tenir l'aplicació.

- Capítol 5, Implementació de l'Aplicació, es detalla com s'ha traduït el que s'ha dissenyat al llenguatge de programació concret i com s'ha creat el codi per a comunicar amb els diferents serveis triats.
- Capítol 6, Conclusions, es fa un anàlisi de com s'han assolit els objectius inicials i si hi ha hagut molta desviació del que s'ha proposat inicialment amb el producte final. També es recull les conclusions personals sobre la geolocalització dels equips a través de la IP dels equips.

2. Marc teòric

2.1. Encapsulament i la pila TCP/IP

Les comunicacions a través de la xarxa es realitzen mitjançant paquets formats per seqüències d'uns i zeros, és a dir, de codi binari. Aquests paquets són la unitat bàsica d'aquestes comunicacions i estan formats per un encapçalament on es troba la direcció d'enviament i la de recepció i una part de càrrega on s'inclouen les dades que es volen transmetre.

La pila de protocol TCP/IP és un conjunt de protocols que estableix com s'ha d'organitzar la informació dintre d'aquests paquets, es tracta d'una organització per capes que dintre del paquet s'organitzen introduint la capa superior dintre de la capa inferior següent, aquest sistema s'anomena encapsulament.

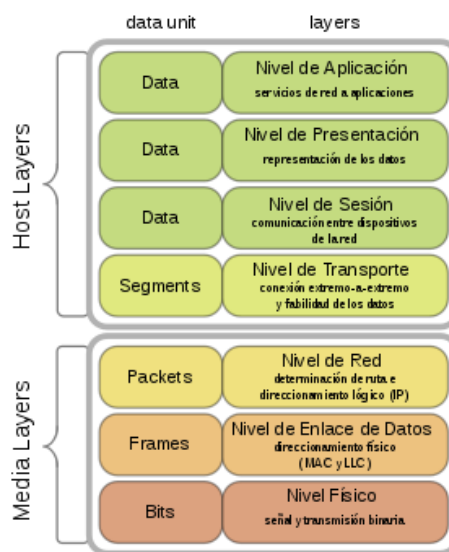
Les capes que formen part de la pila TCP/IP són les següents:

- Capa d'aplicació, aquesta correspon a la part superior de la pila i defineix la capa de l'aplicació, aquesta dona format a un paquet d'informació que es vol transmetre a un altre host. Es on es xifren i es comprimeixen les dades amb el format correcte per a la seva transmissió.
- Capa de transport, aquesta rep el paquet bàsic de la capa d'aplicació i s'encarrega de l'encapsulació per a transmetre el paquet amb el protocol triat en unitats de comunicació. Aquestes unitats de comunicació s'anomenen segments o datagrames. Els protocols utilitzats poden ser UDP o TCP, segons la necessitat de la comunicació, i identifiquen els ports lògics de destí i origen dels paquets per tal que es comuniquin amb l'aplicació correcta.
- Capa de xarxa o d'internet, encapsula el paquet resultant de la capa de transport en unitats anomenades paquets, és l'encarregada d'afegir la informació de xarxa necessària per tal que el paquet arribi del host origen al host destí. En aquest nivell de capa treballen els encaminadors de la xarxa que compleixen la funció de triar el millor camí per a cada paquet per arribar al seu destí. S'afegeix una capçalera IP amb les dades de l'origen i el destí, es crea el datagrama IP.
- Capa d'accés al medi, en aquest nivell s'encapsula el datagrama IP afegint l'adreça física dels equips de destí i origen. Aquesta adreça física és l'adreça MAC (*Media Access Control*), que es una sèrie de números hexadecimals única per a cada dispositiu de xarxa que l'identifica dels altres equips. El paquet resultant és una estructura on es troben totes les capes estructurades i preparat per a enviar a través del medi físic com pot ser un cable de comunicacions.
- Capa física, on el maquinari acaba d'encapsular el paquet en el format adient per a realitzar la transmissió a través del medi físic utilitzat.



Il·lustració 2. Pila TCP/IP

Aquests nivells de la pila TCP/IP es corresponen amb un altre estàndard també molt utilitzat per definir l'encapsulament de les dades i que divideix aquestes capes en 7 nivells, anomenat estàndard OSI. Les capes de Presentació, Sessió, i Transport es corresponen a la capa de Transport del model TCP/IP. A continuació es mostra l'estàndard OSI:



Il·lustració 3. Estàndard OSI

2.2. Adreçament IPv4

Com s'ha vist en l'apartat anterior, els equips es troben connectats a la xarxa mitjançant interfícies enllaçades entre elles amb equips de xarxa que són els encaminadors i els commutadors, per tal de identificar les interfícies de cada equip s'utilitza el protocol IP, el qual permet assignar a cada interfície un número format per quatre grups de 8 bits (4 bytes). Les adreces IP es divideixen en dues parts, la que identifica la xarxa i la que identifica els equips, i aquestes varien segons la mascara de xarxa com es veurà una mica més endavant. A nivell de maquinari aquestes adreces estan relacionades amb l'adreça física de la interfície, que és única per a cada interfície. Normalment, aquestes adreces IP es mostren en nombres decimals

separats per un punt, i hi ha aproximadament 4 bilions d'adreces possibles. Com es pot veure, aquest numero és insuficient per tal d'adreçar tots els equips que es troben connectats a internet, encara que en el moment de la creació d'Internet es pensava que n'hi hauria prou.

Les adreces IP s'agrupen en xarxes, les quals es defineixen a partir de la mascara de xarxa, aquesta determina quants equips hi ha en una xarxa i també està formada per quatre bytes. Encara que el funcionament d'aquests és diferent a les adreces IP, segons el número de bits identificats amb 0 es pot saber quins bits de l'adreça IP poden variar i per tant quantes adreces IP poden formar la subxarxa corresponent. Per exemple, per a una xarxa identificada amb l'adreça 192.168.1.0, si la mascara de subxarxa és 255.255.255.0 (11111111.11111111.11111111.00000000) significarà que només variaran els 8 últims bits de l'adreçament, i per tant els equips només podran tenir el rang comprés entre 192.168.1.1 al 192.168.1.255 i els equips que estiguin fora d'aquest seran inaccessibles sense un encaminament. Igualment, es pot apreciar que la part de l'adreça IP que identificarà la xarxa seran els 24 primers bits i els hosts s'identificaran amb els 8 últims bits. Altra manera de referenciar una xarxa es fa indicant quants bits de la mascara de xarxa es troben amb el valor 1, així doncs, la xarxa anterior es pot indicar com 192.168.1.0/24. Aquest mecanisme de xarxes és molt important a l'hora d'aprofitar les assignacions de rang d'adreces IP, ja que permet fer subxarxes a partir d'una xarxa concreta, seguint amb l'exemple anterior, dintre de la xarxa 192.168.1.0/24 es podria dividir en dos subxarxes indicades com 192.168.1.0/25 i 192.168.1.64/25.

Altre concepte important en l'adreçament IP és la diferència entre adreça IP pública i privada, com ja s'ha dit al principi, els equips estan connectats entre ells amb encaminadors i commutadors, els encaminadors són els encarregats de redirigir els paquets destinats a equips ubicats fora de la xarxa. L'adreça IP que els equips tenen dintre de la xarxa privada s'anomena adreça privada i només és visible per als equips que es troben en la mateixa xarxa, quan aquests equips envien paquets a equips situats fora de la xarxa privada, l'encaminador és l'encarregat de substituir l'adreça privada per l'adreça que ell té dintre de la xarxa pública, de tal manera que les respostes a aquest paquet aniran primer dirigides a l'adreça de l'encaminador, per posteriorment, ser substituïda per l'adreça de l'equip remitent i arribar la resposta al seu destinatari. A partir d'aquí es pot veure que els equips d'una xarxa privada, sigui tant gran com sigui, només tindran una adreça pública (o el número d'adreces IP públiques que tinguin assignades pel ISP corresponent). Aquest mecanisme d'adreçament permet l'assignació d'un gran número d'adreces IP que d'altra manera no seria possible al estar aquestes repetides.

Finalment, les adreces IP es divideixen en grups anomenats Classes, on cada grup defineix un rang d'adreces concret i per convenció estan destinats a uns objectius concrets, a continuació es mostren les diferents Classes:

Classe	Rang	Objectiu
A	1.0.0.0 a 126.255.255.255	Reservada per a xarxes molt grans. Les adreces del rang 127.0.0.0 s'utilitzen per identificar l'equip local.
B	128.0.0.0 a 191.255.255.255	S'utilitzen per a xarxes de mida mitjana.
C	192.0.0.0 a 223.255.255.255	Reservada per a xarxes petites o mitjanes.
D	224.0.0.0 a 239.255.255.255	Utilitzat per a comunicacions <i>multicast</i> .
E	240.0.0.0 a 255.255.255.255	Reservada amb propòsits experimentals

Taula 2. Classes IP

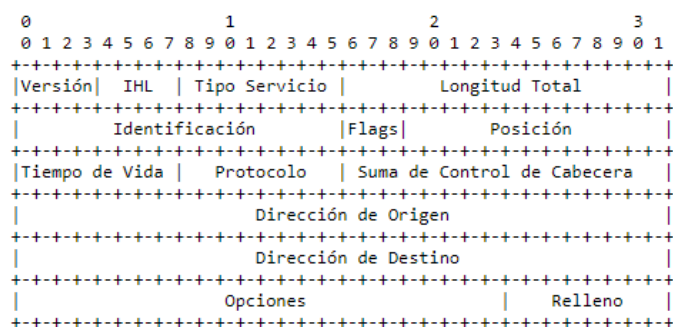
Dintre de les diferents classes es diferencien adreces per a xarxes privades o per xarxes públiques, els rangs d'adreces reservats per adreces privades són els següents:

10.0.0.0 a 10.255.255.255
172.16.0.0 a 172.16.255.255
192.168.0.0 a 192.168.255.255
169.254.0.0 a 169.254.255.255

Taula 3. Rangs d'adreces privades

2.3. Protocol ICMP (Protocol de Missatges de Control d'Internet)

El protocol ICMP està situat dintre del nivell de paquets o també conegut com a nivell de xarxa o IP, aquest protocol s'utilitza per tal de detectar possibles errors dintre dels datagrames IP. Les especificacions d'aquest protocol estan recollides dintre de la norma RFC 792, on s'indica que el protocol ICMP consta de diferents missatges que possibiliten diagnosticar els problemes en el processament dels datagrames. A continuació, es mostra l'estructura d'un datagrama IP.



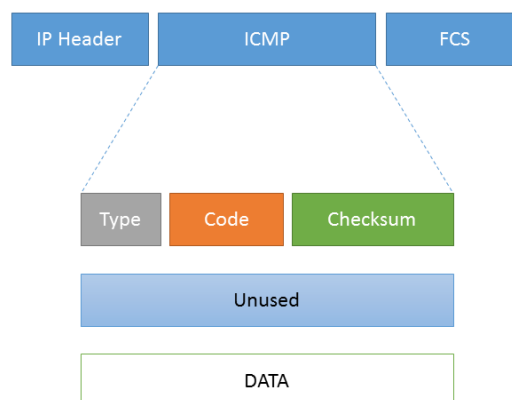
Il·lustració 4. Estructura datagrama IP

Dintre de la capçalera trobem els camps següents:

- Versió, el qual indica si es tracta de protocol IPv4 o IPv6.
- IHL, longitud de la capçalera sense comptar les dades.
- Tipus de servei, es refereix a la qualitat del servei indicada per al datagrama, els possibles valors es recullen en l'especificació RFC 1349.
- Longitud total, és la longitud del datagrama comptant les dades, s'expressa en bytes.

- Identificació, és un número únic que dona suport al receptor per tal de tornar a unir els paquets si hi ha fragmentació.
- *Flags*, és tracta de valors de control de la fragmentació.
- Posició, indica la posició del paquet si hi ha fragmentació per ajudar a unir els paquets.
- Temps de vida (TTL o *time to live*), aquest camp indica el nombre màxim de salts que pot fer el paquet abans d'arribar al destí. Per cada node que passa el paquet, es descompta una unitat i si el valor de TTL arriba a 0 abans de trobar el host indicat, el node que fica el valor a 0 retorna un missatge ICMP indicat que s'ha excedit el TTL.
- Protocol, indica el protocol de nivell superior al qual el protocol IP ha d'entregar les dades. En el cas del protocol ICMP el valor ha de ser 1.
- Suma de control de capçalera, aquest valor es calcula fent una suma de bytes de capçalera i permet comprovar si el paquet s'ha deteriorat durant la transmissió.
- Direcció d'origen, indica la direcció IP del host emissor del paquet.
- Direcció de destí, indica la direcció del host destinatari del paquet.
- Opcions, camp de longitud variable que s'utilitza per a la depuració o proves de xarxa.

A continuació de la capçalera IP trobem el datagrama del protocol ICMP:



Il·lustració 5. Capçalera datagrama IP del protocol ICMP

Mitjançant la combinació dels camps *type* i *code* es poden detectar diferents tipus d'errors i quin és el motiu pel qual s'han produït. El primer camp, *type*, indica quin tipus de missatge ICMP s'està produint, en el nostre projecte ens centrarem en el *type* 3, el qual es refereix a que no es pot arribar al equip indicat (*Unreachable host*).

El camp *code* ens indica quin és el motiu pel qual no es pot arribar al destí, a continuació es mostren els possibles combinacions *Type* i *Code* que recull aquest protocol.

ICMP Type	Code	Description
0	0	<i>echo reply</i>
3	0	<i>destination network unreachable</i>
3	1	<i>destination host unreachable</i>
3	2	<i>destination protocol unreachable</i>
3	3	<i>destination port unreachable</i>
3	6	<i>destination network unknown</i>
3	7	<i>destination host unknown</i>
4	0	<i>sourcequench (congestion control)</i>
8	0	<i>echo request</i>
9	0	<i>router advertisement</i>
10	0	<i>router discovery</i>
11	0	<i>TTL expired</i>
12	0	<i>IP header bad</i>

Taula 4. Respostes ICMP

El camp *checksum* s'utilitza per a controlar que les dades del datagrama ICMP estiguin correctes, si al rebre el paquet el receptor, el càlcul del valor de *checksum* és diferent, aquest paquet es rebutja.

2.4. Ping

Es tracta d'un programa que utilitza les característiques del protocol ICMP per tal de verificar la connectivitat amb un host determinat. Utilitza dos tipus de missatge, *echo reply* i *echo request*. Concretament es crea un datagrama ICMP amb *type 8* i *code 0* (*echo request*), amb la capçalera IP que indica la adreça IP origen, la adreça IP de destí, el protocol amb el valor 1 (ICMP), i el tipus de servei 0 (rutinari). Quan el host de destí rep el paquet, aquest retorna un datagrama ICMP a l'origen amb la combinació *type 0* i *code 0* el qual indica que el paquet s'ha rebut correctament i que el host és assolible.

L'eina *Ping* s'utilitza per tal de comprovar la connectivitat amb altres equips, per a facilitar les comprovacions incorpora alguns paràmetres que permeten modificar les característiques del paquet, a Windows aquestes són algunes d'elles:

- */t*: realitza *ping* fins que es deté manualment.
- */a*: resol la direcció com a nom de host.
- */l*: especifica la mida en bytes del paquet ICMP que s'envia, pot ser com a màxim de 65527 bytes.
- */f*: indica que els paquets ICMP no es poden fragmentar.
- */i*: permet modificar el TTL del paquet ICMP enviat, normalment s'estableix entre 128 i 255 salts.

2.5. Traceroute

Altra aplicació que aprofita les característiques del protocol ICMP combinades amb el paràmetre TTL (*time to live*) de la capçalera IP, per tal de determinar quins salts fan els paquets abans d'arribar al host indicat. El seu objectiu és identificar els equips per on passen els paquets per arribar d'un equip a un altre.

El seu funcionament és el següent, el host d'origen crea un paquet amb un TTL=1 i l'envia, quant aquest arriba al primer node, aquest node resta una unitat al valor TTL, com aquest valor és igual a 0 i encara no ha arribat al host destí, retorna un paquet ICMP amb *type* 11 i *code* 0, el qual indica que el valor TTL ha expirat i l'adreça IP de l'equip a on ha arribat el paquet. Al rebre aquest paquet, el host origen incrementa el valor TTL una unitat i torna a enviar el paquet, si l'equip a on arriba ara amb els dos salts no és el host destí, aquest tornarà a rebre la mateixa resposta amb les dades del següent node. Quan finalment, després d'incrementar el valor TTL fins que sigui el necessari, el paquet arriba al host destí, aquest retorna un paquet *echo replay*. Així l'equip que ha executat el *traceroute* sabrà amb el valor final de TTL quants salts cal per arribar a l'equip destí, i a més haurà pogut recopilar informació dels nodes intermedis entre ells. Un possible resultat mentre es realitza aquest procés és que hi hagi equips que no responguin i excedeixin el temps determinat per a rebre la resposta, això pot ser degut a que aquests equips no tenen activat el protocol ICMP per diferents raons.

2.6. Domain Name System (DNS)

Es tracta d'un sistema jeràrquic descentralitzat que relaciona les adreces IP dels equips connectats a la xarxa, ja sigui pública o privada, amb noms intel·ligibles per a les persones. Aquest servei va sorgir per la dificultat de recordar series de números, com són les adreces IP, que tenen les persones. Per tal de facilitar aquesta tasca aquest servei permet que per accedir a un domini a la xarxa l'usuari només hagi de saber un nom que resulta més fàcil de recordar, com per exemple www.amazon.es.

El sistema DNS es compon per una sèrie de servidors distribuïts per la xarxa que quan reben una consulta DNS de resolució de noms amb el nom d'un domini determinat, aquest servidor consulta si té la resposta a aquesta consulta de resolució, si la té, retorna el valor de l'adreça IP concreta, si no la té, trasllada la consulta a un altre servidor DNS. Normalment, el primer servei DNS l'ofereix el mateix equip, el qual comprova a la seva memòria cau si disposa de la informació demanada, si no és així, passa la consulta a un altre servidor.

2.7. Whois

Whois és un sistema que emmagatzema les dades relacionades amb els noms de dominis proporcionant informació de la propietat d'aquests i qui és el responsable. *Whois* té el seu origen als inicis d'Internet, quan el Grup de Treball de Enginyeria en Internet (IETF) va publicar un protocol per als usuaris de ARPANET. Aquest protocol va ser heretat i mantingut per la ICANN quan es va crear al 1998.

Amb el creixement d'Internet, ICAAN va delegar la gestió d'aquest directori per continents per facilitar l'accés i la operativitat d'aquest, passant de tenir un únic directori a tenir cinc, aquests es coneixen com a Registres Regionals d'Internet (RIR). Les directrius del repartiment de les adreces d'Internet en continents i per a usos especials està recollida a les directrius RFC 1466 i RFC 7249, en la primera s'especifica els rangs d'adreces que pertany a cada bloc. Els servidors que s'encarreguen de donar suport a les cerques de tipus *Whois* en cada continent són:

Nom	Adreça d'Internet	Zona geogràfica
American Register for Internet Numbers (ARIN)	whois.arin.net	Amèrica Anglosaxona
RIPE Network Coordination Centre (RIPE NCC)	whois.ripe.net	Europa, Orient mitjà, i Àsia Central
Asia-Pacific Network Information Centre (APNIC)	whois.apnic.net	Àsia i la Regió Pacífica
Latin American and Caribbean Internet Address Registry (LACNIC)	whois.lacnic.net	Amèrica Llatina i el Carib
African Network Information Centre (AfriNIC)	whois.afrinic.net	Àfrica

Taula 5. Servidors Whois dels diferents continents

Com es pot deduir, les adreces IP públiques dels equips poden ser una aproximació a la geolocalització, situant a aquests en un continent determinat. Però, cal tenir en compte que el servei *Whois* només indica la titularitat d'un domini, el que vol dir que un equip amb una determinada adreça IP, que per la divisió de rangs IP es troba en continent, pot indicar que el titular d'aquesta i per tant les dades de contacte estan en un altre. Així doncs, no es pot considerar un mètode gaire fiable de geolocalització, ja que tampoc és la seva funció.

El que si és important a l'hora de treballar amb aquest servei és saber de quin continent és l'adreça IP, així es pot fer la consulta directament al servidor del continent concret.

L'estructura del servei proporcionat per la IANA és jeràrquica, els usuaris tenen adreces IP assignades pels ISP, que a la vegada obtenen el rang d'adreces assignat d'una autoritat LIR (*Local Internet Registry*) o NIR (*National Internet Registry*), aquestes autoritats igualment obtenen l'assignació del rang d'adreces IP del seu RIR corresponent, que com ja hem dit es divideixen per continents.

3. Anàlisi de recursos

3.1. Geolocalització IP

La geolocalització IP es refereix a la possibilitat d'identificar la ubicació geogràfica d'un determinat equip a través de la seva adreça IP. L'objectiu final és determinar la ubicació indicant el país, la ciutat, etc. Igualment, amb la geolocalització es relaciona el concepte de geocodificació, que es refereix a relacionar l'adreça IP a una latitud i longitud geogràfica per a situar l'equip geogràficament.

Com s'ha vist anteriorment, l'assignació d'adreces IP és un procés jeràrquic que va des d'una autoritat global, que és la IANA, fins als ISP locals, els quals assignen les adreces concretes als seus clients. Són aquests ISP els que en principi tenen la informació concreta de a quina ubicació correspon una adreça IP. Així, doncs, la geolocalització dels equips es fa mitjançant la consulta de dades proporcionades pels ISP. Aquestes dades estan emmagatzemades en diferents bases de dades que poden ser de pagament o gratuïtes, i tenen una precisió variable (entre un 50% o un 98%).

També existeix un protocol experimental que es va dissenyar al 1996 i que es recull en la norma RFC 1876 anomenat DNS-LOC, el qual relaciona la localització geogràfica amb el nom de domini. Aquesta informació es troba en els servidors DNS que l'implementen, encara que sembla que no ha arribat a ser un sistema gaire utilitzat.

La utilitat de la geolocalització és diversa, habitualment s'utilitza amb les següents finalitats:

- Seguretat, moltes vegades els servidors reben atacs DDoS des de determinades zones geogràfiques, la possibilitat de detectar aquestes zones permet denegar l'accés al servidor des d'aquestes zones sense deixar sense servei a la resta, mentre es soluciona el problema de seguretat.
- Comercial, per a un negoci en xarxa pot ser una dada estratègica identificar des de quina zona geogràfica està accedint un usuari. Aquesta informació pot determinar quins productes tindran més interès per al comprador, i així el venedor pot modificar les ofertes segons la geolocalització del comprador.

Així mateix, existeixen comerços a la xarxa que no proporcionen servei a determinades zones geogràfiques, ja sigui per falta de distribució o per tractar-se de zones conflictives.

A continuació es mostren algunes de les bases de dades de geolocalització IP més conegudes.

3.1.1. MaxMind

És una de les empreses més coneguda dedicada a la geolocalització, proveeix bases de dades amb informació de geolocalització IP, ofereix dos tipus de productes, un de pagament, anomenat GeoIP, i un de gratuït, anomenat GeoLite2 (d'aquesta opció només donaran suport fins abril de 2018 i al 2019 ja no hi haurà la informació de la latitud i la longitud). Ambdues són

bases de dades amb informació de les adreces IP relacionades amb l'adreça geogràfica. La diferència és el grau d'exactitud i el suport que es dona al client. Les consultes a la base de dades es poden fer tant en línia a través d'una API com descarregant les dades i emmagatzemant-les en una base de dades local, per aquesta segona opció es dona suport per a tenir la base de dades actualitzada amb descarregues periòdiques. El preu de la base de dades GeoIP canvia segons es vulgui una precisió a nivell de país o de ciutat. A la imatge següent es mostren les tarifes d'aquest servei.

Database	Country	City
Application	Segment your visitors by country	When more granular data is needed
Continent	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Country	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Regional Subdivision		<input checked="" type="checkbox"/>
City		<input checked="" type="checkbox"/>
Postal Code		<input checked="" type="checkbox"/>
Approximate Latitude/Longitude*		<input checked="" type="checkbox"/>
Update Frequency	Weekly	Weekly
Site License Fee (Includes one Month of updates)	\$50	\$370
Monthly Billing	\$24 per month	\$100 per month
Annual Billing	\$314 for the first year and \$288 for each year thereafter.	\$1470 for the first year and \$1200 for each year thereafter.
	More Info	More Info

* Latitude and longitude are not precise and should not be used to identify a particular street address or household.

Il·lustració 6. Tarifes MaxMind

3.1.2. IP2Location

També es tracta d'una empresa dedicada a oferir serveis de geolocalització, al igual que *MaxMind*, ofereix la possibilitat d'accedir a través d'API, descarregant una base de dades, o amb un servei d'emmagatzematge en el núvol. Ofereix una ampla varietat de bases de dades on en cada una s'incrementa la informació, per exemple la base de dades BD1 ofereix el país segons l'adreça IP i la BD24 ofereix el país, regió, ciutat, latitud, longitud, codi postal, zona horària, ISP, domini, velocitat de xarxa, codi d'àrea, el temps, mòbil, elevació, i ús de la base de dades. Depenent del nombre de dades que es vol el preu variarà, tal com es pot apreciar a continuació.

PRODUCT	YEARLY	QUANTITY
IP1 IP-Country Database	\$49.00	1
IP2 IP-Country-ISP Database	\$199.00	1
IP3 IP-Country-Region-City Database	\$199.00	1
IP4 IP-Country-Region-City-ISP Database	\$349.00	1
IP5 IP-Country-Region-City-Latitude-Longitude Database	\$349.00	1
IP6 IP-Country-Region-City-Latitude-Longitude-ISP Database	\$499.00	1

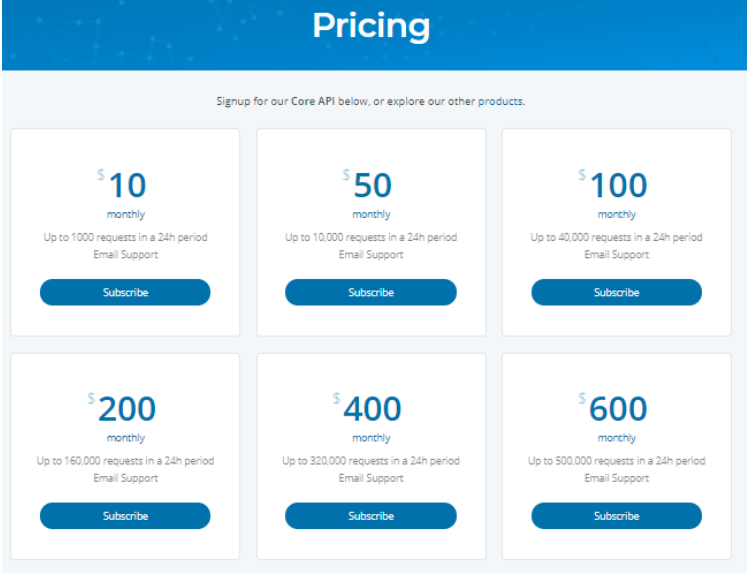
Il·lustració 7. Tarifes IP2Location

3.1.3. *Ipinfo.io*

Es una opció que ofereix consultes a través del seu API, similar a les altres dues però molt més senzilla, és gratuïta si es fan menys de 1000 consultes diàries. Aquesta opció pot ser interessant per a solucions que no realitzin moltes consultes. Realitzar consultes és molt senzill, només cal introduir l'adreça web *ipinfo.io* amb l'adreça IP que es vol consultar a continuació:

ipinfo.io/8.8.8.8

El retorn de la consulta és la mateixa adreça IP, el *hostname*, localització, organització, ciutat, regió, país, i telèfon si hi ha. Es pot especificar que es vol rebre la resposta en format JSON, i també dona la possibilitat de realitzar consultes de camps concrets com pot ser ciutat o país. Els preus si s'excedeixen les 1000 consultes diàries són:



The screenshot shows a pricing table with a blue header titled "Pricing". Below the header, there is a sub-header: "Signup for our Core API below, or explore our other products." The table contains six pricing cards arranged in two rows of three. Each card displays a price in dollars per month, the number of requests allowed in a 24-hour period, and the inclusion of email support. Each card has a "Subscribe" button at the bottom.

Price (Monthly)	Requests (24h)	Support
\$ 10	Up to 1000	Email Support
\$ 50	Up to 10,000	Email Support
\$ 100	Up to 40,000	Email Support
\$ 200	Up to 160,000	Email Support
\$ 400	Up to 320,000	Email Support
\$ 600	Up to 500,000	Email Support

Il·lustració 8. Tarifes *Ipinfo.io*

3.1.4. *Ip-api*

Finalment, altra API que proporciona les dades sobre la geolocalització d'un host és *Ip-api.com*, es tracta d'un servei gratuït, que permet fer consultes molt complertes sobre dades de geolocalització. Per defecte retorna els camps següents:

Nom	Descripció	Exemple	Tipus
status	always success	<i>success</i>	string
country	country	<i>United States</i>	string
countryCode	country short	<i>US</i>	string
region	region/state short	<i>CA or 10</i>	string
regionName	region/state	<i>California</i>	string
city	city	<i>Mountain View</i>	string
zip	zip code	<i>94043</i>	string
lat	latitude	<i>37.4192</i>	float
lon	longitude	<i>-122.0574</i>	float
timezone	city timezone	<i>America/Los_Angeles</i>	string
isp	ISP name	<i>Google</i>	string
org	Organization name	<i>Google</i>	string
as	AS number and name, separated by space	<i>AS15169 Google Inc.</i>	string
reverse	Reverse DNS of the IP	<i>wi-in-f94.1e100.net</i>	string
mobile	mobile (cellular) connection	<i>true</i>	bool
proxy	proxy (anonymous)	<i>true</i>	bool
query	IP used for the query	<i>173.194.67.94</i>	string

Taula 6. Taula de camps de IP-API

I si hi cap error el format de missatges d'error següent:

Nom	Descripció	Exemple	Tipus
status	always fail	<i>fail</i>	string
message	error message	<i>reserved range</i>	string
query	IP used for the query	<i>173.194.67.94</i>	string

Taula 7. Missatges d'error de Ip-api

També proporciona la possibilitat de personalitzar la consulta mostrant només els camps que es necessitin, això es fa indicant els camps desitjats a la pàgina web i aquesta ens indica un número que cal afegir a la consulta, tal com es mostra a continuació:

Selectable output

If you don't require all the returned fields, use the GET parameter **fields** to specify which fields to return.

Separate the fields by comma (example: `fields=status,message,query,city`) or use a generated, numeric value (to save bandwidth).

Field generator

Tip: disabling **reverse** may improve performance

name	X
country	<input checked="" type="checkbox"/>
countryCode	<input type="checkbox"/>
region	<input type="checkbox"/>
regionName	<input checked="" type="checkbox"/>
city	<input checked="" type="checkbox"/>
zip	<input type="checkbox"/>
lat	<input checked="" type="checkbox"/>
lon	<input checked="" type="checkbox"/>
timezone	<input type="checkbox"/>
isp	<input type="checkbox"/>
org	<input type="checkbox"/>
as	<input checked="" type="checkbox"/>
reverse	<input type="checkbox"/>
mobile	<input type="checkbox"/>
proxy	<input type="checkbox"/>
query	<input checked="" type="checkbox"/>
status	<input checked="" type="checkbox"/>
message	<input checked="" type="checkbox"/>

Human readable

```
country,regionName,city,lat,lon,query,status,message
```

Example:

```
http://ip-api.com/xml/?fields=country,regionName,city,lat,lon,query,status,message
http://ip-api.com/xml/wikipedia.org?fields=country,regionName,city,lat,lon,query,status,message
```

Numeric

```
57561
```

Example:

```
http://ip-api.com/xml/?fields=57561
http://ip-api.com/xml/wikipedia.org?fields=57561
```

Il·lustració 9. Pantalla de selecció de camps

La consulta es realitzarà mitjançant la següent adreça web:

<http://ip-api.com/csv/8.8.8.8?fields=57561>

I la resposta a la consulta és la següent:

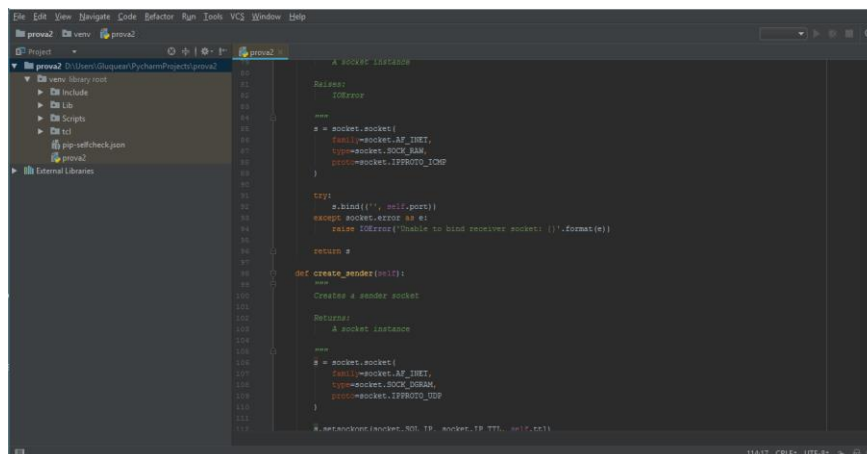
```
success,"United States",California,"Mountain View",37.4229,-122.085,8.8.8.8
```

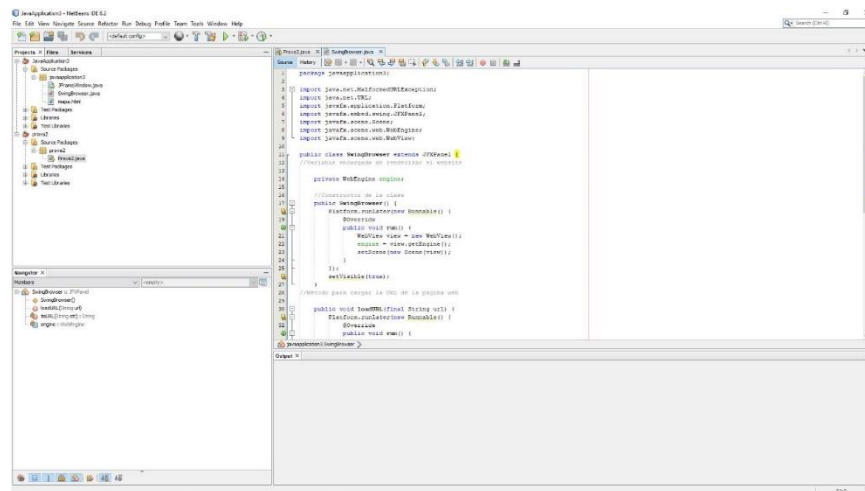
3.2. Llenguatges de programació

Per crear el producte es vol fer una aplicació basada en finestres i que trobi la ruta entre l'equip local i un domini mitjançant el protocol ICMP. Per crear l'aplicació s'ha avaluat diferents llenguatges de programació que es mostren a continuació.

3.2.1. Python

Es tracta d'un llenguatge de programació que permet la programació orientada a objectes i a més la creació d'aplicacions basades en finestres. S'ha utilitzat el IDE *pyCharm* per avaluar les possibilitats que ofereix aquest llenguatge.





Il·lustració 11. Entorn IDE Netbeans 8.2

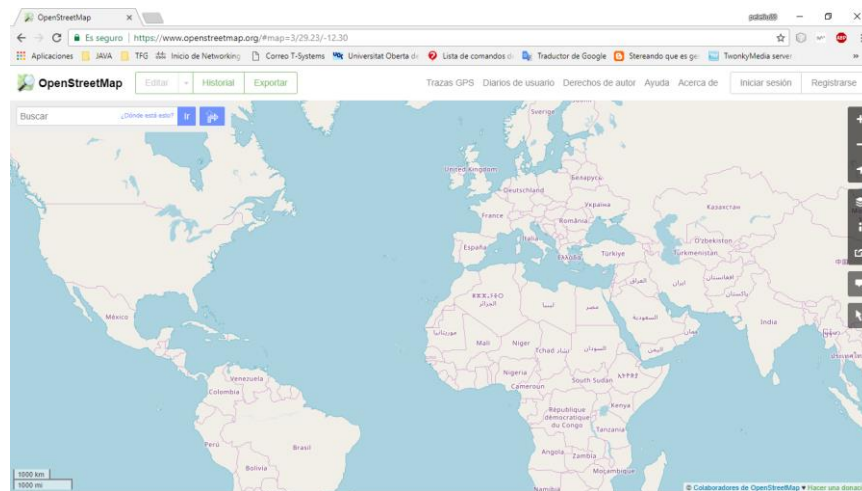
3.3. Serveis de Mapes

Altra característica del producte que es vol crear és la possibilitat de mostrar els diferents nodes trobats a la ruta dels paquets en un mapa identificant la ruta que segueixen aquests. Per aquest objectiu s'han buscat serveis a la xarxa que proporcionin les funcionalitats desitjades.

3.3.1. Open Street Map API

Es tracta d'una solució *Open Source* la qual està elaborada pels mateixos usuaris que creen els mapes de manera col·laborativa. La informació geogràfica es captura amb GPS, mòbils, i altres fonts lliures i aquesta es emmagatzemada en una base de dades sota llicència ODBL (*Open Database License*).

Ofereix una API molt completa a l'hora de manipular els mapes i informar sobre noves dades per afegir-les al mapa. Pel que fa a crear marques i unir-les per a mostrar les rutes, no es troba una manera gaire eficient de fer-ho. L'objectiu de l'API és més per afegir dades, tal com ja s'ha dit. Si es vol treballar amb aquest servei de mapes cal recórrer a llibreries de tercers.



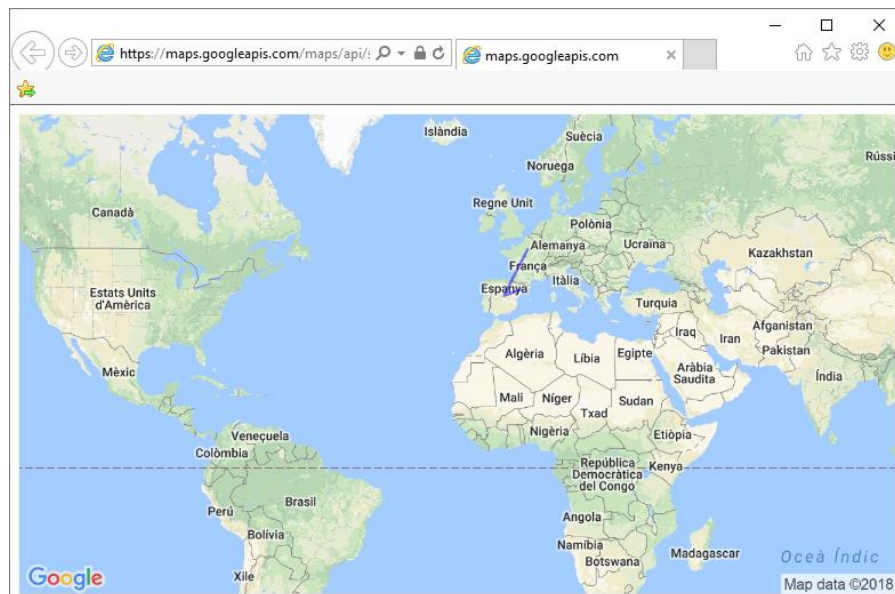
Il·lustració 12. Open Street Map

3.3.2. Google Maps API

Forma part del servei *Google Maps*, és una solució que permet la seva utilització gratuïta amb un número limitat de consultes. Ofereix diferents API depenent de el que es vol implementar amb aquestes. Per poder utilitzar les API l'usuari s'ha de registrar i indicar quina API vol utilitzar i el nom del projecte. Per a cada tipus d'API *Google* proporciona a l'usuari registrat una clau (*key*), la qual s'ha d'afegir a la consulta per que *Google Maps* respongui.

En comparació amb la solució anterior, aquesta proporciona més versatilitat al oferir diferents opcions més ajustades a les necessitats de l'usuari. En el cas del producte que es vol realitzar hi ha dos API que poden ajustar-se als requisits previs.

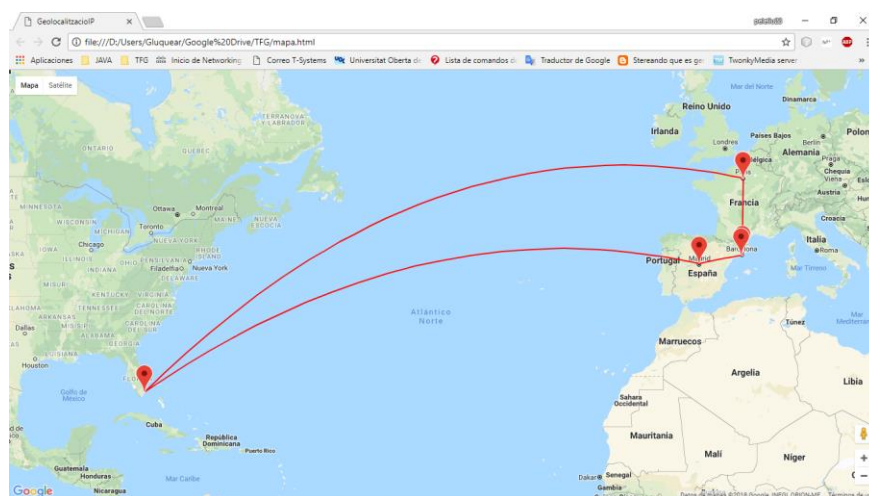
La primera opció és *Google Static Maps*, aquesta és una solució web la qual, mitjançant una consulta amb una URL que conté tots els paràmetres desitjats, retorna una imatge en format *jpg* o *png* on es mostra el mapa demanat. Per exemple, amb la consulta https://maps.googleapis.com/maps/api/staticmap?center=Gran%20Canaria&zoom=2&size=640x350&scale=2&path=color:0x0000ff|weight:2|Lleida|Madrid|Paris&key=AlzaSyD_V_IRBGpyWQWJzpp8yEZlwLTfCbEIBso, l'API retorna la imatge del mapa següent:



Il·lustració 13. Google Static Maps API

Els inconvenients que presenta aquesta API són que es limita la resolució màxima del mapa retornat a 640x480 píxels i si es vol una millor resolució s'ha de contractar un pla de pagament, aquesta és una resolució molt petita si es vol comprovar on està situat exactament cada marca. I que es tracta d'una imatge estàtica amb la qual no es pot interactuar, és a dir, no es pot ampliar la imatge o desplaçar-la.

L'altra opció que pot complir amb les necessitats del projecte és *Google Maps JavaScript API*, aquesta també és una solució web però la consulta es fa a través de *JavaScript* i el que es retorna és una pàgina web dinàmica que es pot inserir en el projecte i permet interactuar amb aquesta. D'aquesta manera, generant una pàgina web s'obté un mapa amb la informació desitjada que es pot manipular.



Il·lustració 14. Google Maps JavaScript API

En aquesta segona opció no hi ha límit de resolució i també és gratuïta si no s'excedeix el nombre màxim de consultes. A continuació, es pot veure el detall de tarifes per aquestes opcions.

Web	ESTÁNDAR	PREMIUM
Google Maps JavaScript API	Gratis hasta 25 000 cargas de mapa por día. ³	Precio basado en el volumen requerido.
Google Static Maps API	USD 0,50 cada 1000 cargas de mapa adicionales, hasta 100 000 diarias, si está habilitada la facturación.	Funciones premium mejoradas: <ul style="list-style-type: none"> • Sin publicidad, garantizado. • Tamaño de imagen de hasta 2048 x 2048 píxeles.
Google Street View Image API		Para obtener más información, consulta Premium Plan Límites e índices de utilización .
Google Maps Embed API	Uso gratuito ilimitado.	---

Il·lustració 15. Tarifes Google Maps JavaScript

3.4. Elecció de recursos

Primerament, s'ha realitzat l'elecció del llenguatge de programació, ja que la resta de recursos hauran de poder utilitzar-se amb aquest. Després de comparar les possibilitats que ofereixen els tres llenguatges analitzats, es pot veure que amb qualsevol es pot implementar un programa que executi un *traceroute*, però a l'hora de fer el programa s'ha trobat més intuïtiu amb Java utilitzant la llibreria *JPCap*, amb aquesta llibreria es pot capturar els paquets ICMP retornats amb facilitat, i a més, aquesta llibreria és un bon suport per a identificar els paquets. Altra avantatge que proporciona Java davant els altres llenguatges, és lo complerts que són els IDE's principals d'aquest llenguatge, *Eclipse* i *Netbeans*. Ambdós ofereixen eines molt complertes per crear aplicacions basades en finestres, fent molt àgil la creació d'aquestes mitjançant les llibreries gràfiques *Swing*. Així doncs, el llenguatge de programació triat és **Java**, i concretament s'utilitzarà l'IDE *Netbeans* de Oracle per a implementar el producte.

En segon lloc, una vegada triat el llenguatge de programació, s'ha avaluat quin recurs de xarxa s'utilitzarà per identificar la latitud i la longitud de cada node. En aquest cas, una primera elecció és *MaxMind* per ser un dels productes més utilitzats en el camp de la geolocalització IP, aquest proporciona dos tipus de mètodes de treball, un *online* de pagament que proporciona les dades utilitzant un API de tercers, i un altre *offline* que funciona descarregant-se les dades en diferents arxius i després realitzant les consultes en local a aquests arxius. Aquesta segona opció planteja dos problemes, el primer i més important, és que el conjunt d'arxius de una mida de més de 300 Mbytes, la segona és que no es dona suport ni actualització d'aquests arxius a partir de l'abril de 2018. L'altra opció, *IP2Location* també ofereix un mètode *offline* descarregant un fitxer en format CSV amb les dades, però és de pagament i si es vol obtenir la latitud i longitud el preu és massa elevat. D'aquesta manera queden descartades aquestes dos opcions encara que semblen les més utilitzades. Entre les opcions gratuïtes online, hi ha una que destaca més que l'altra, aquesta és *ip-api*, aquesta ofereix el mateix servei que *ipinfo*, però amb les proves realitzades té més precisió, a més, mitjançant el format de selecció de consulta, permet consultar només les dades que es necessiten, fent més fàcil la manipulació d'aquestes. És per això que, finalment s'ha optat per triar ***ip-api*** com a servei d'informació de la geolocalització dels nodes.

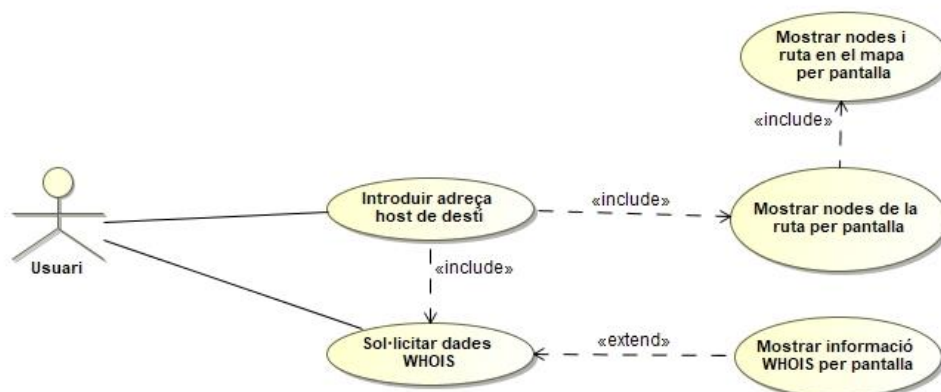
Per últim, cal seleccionar quin servei de mapes s'utilitzarà per a mostrar la informació dels nodes. En aquest cas, s'han avaluat dos serveis, però el que ha mostrat ser més complet ha estat *Google Maps*. En un principi s'han realitzat proves amb *Google Static Maps*, el qual ofereix totes les funcionalitats que es volen implementar, però quan es vol redimensionar o ampliar la imatge aquest no dona la possibilitat, fent molt difícil distingir els nodes quan aquests es torben en una mateixa ciutat. L'altra opció que ofereix *Google Maps*, és *Google Maps JavaScript*, amb aquesta opció es pot generar una pàgina web que es pot inserir en l'aplicació mitjançant Java Swing i permet redimensionar el mapa, fer zoom de la zona que es vol veure amb més detall, o canviar el format del mapa a vista satèl·lit si es prefereix. Finalment el servei de mapes triat és ***Google Maps JavaScript***.

4. Disseny de l'aplicació

4.1. Cas d'ús

L'aplicació a de permetre que l'usuari intrueixi un domini o adreça IP i aquesta retorni en primer lloc els nodes que hi ha entre l'equip origen i el domini indicat. Una vegada acceptat el domini l'aplicació generarà el llistat de nodes i al finalitzar mostrarà automàticament aquests i la ruta en un mapa, per tal de ubicar-los geogràficament.

Altra funcionalitat que ha de proporcionar l'aplicació és la presentació de les dades corresponents al domini demanat que estan emmagatzemades en el servidor *Whois* corresponent. Aquesta funció serà independent de la de realitzar la traça i mostra de dades en el mapa, i es podrà realitzar en qualsevol moment del procés.



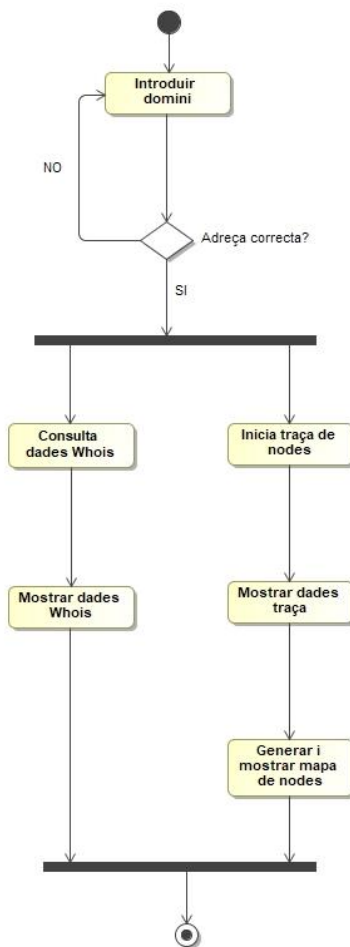
Il·lustració 16. Cas d'us

4.2. Diagrama d'Estats

Els estats que tindrà l'aplicació seran els que es detallen a continuació:

- L'usuari introduirà l'adreça del domini a consultar.
- Si l'adreça no és correcta, l'aplicació ha de indicar-ho i demanar que es torni a ficar una adreça vàlida.
- Si l'adreça és vàlida, es podran realitzar dues accions independents.
- Si es demana l'inici de la traça, s'inicia aquesta per determinar els nodes que hi ha entre l'equip local i el domini.
- Es mostren les dades dels nodes trobats amb les dades necessàries.
- Es genera i es mostra el mapa amb la informació.

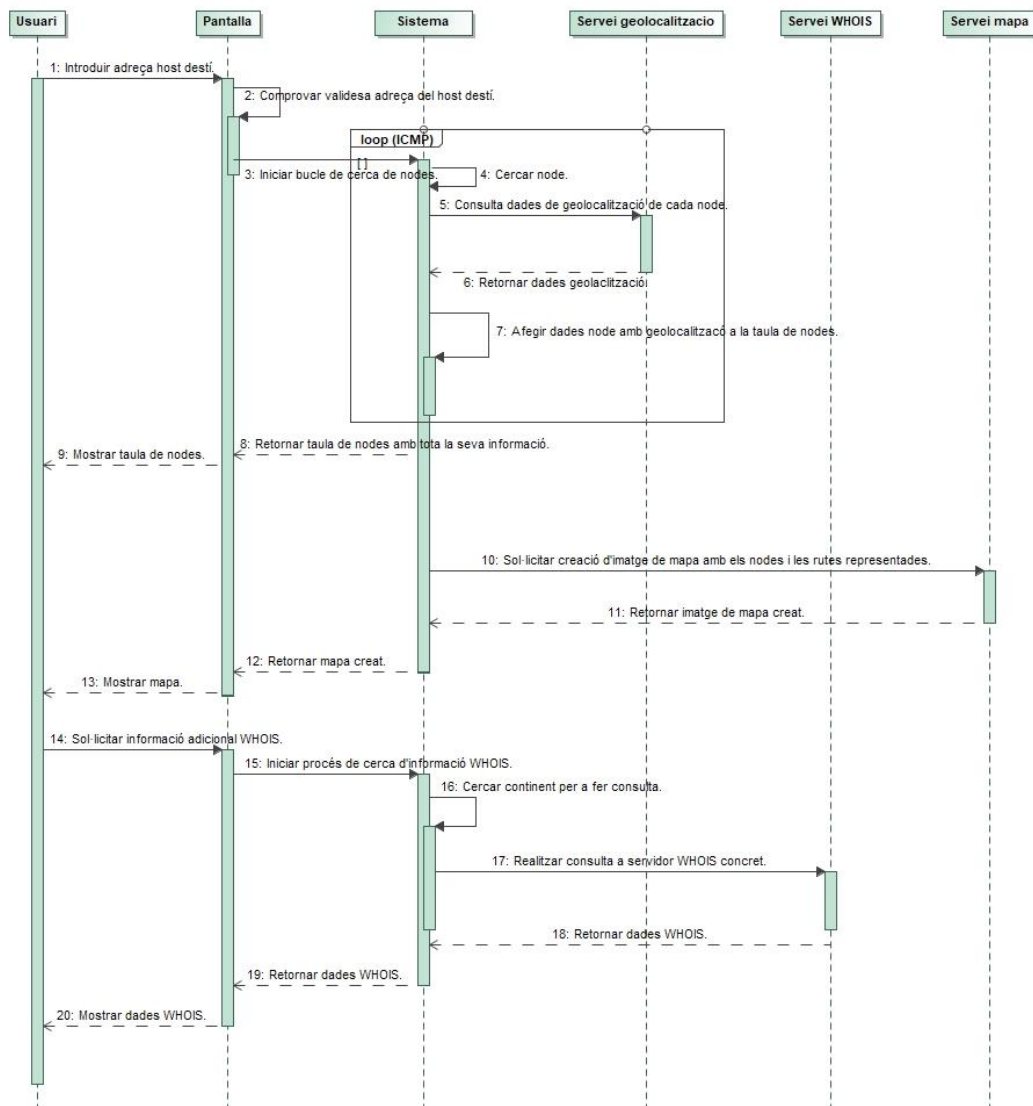
- Si es tria la consulta de la informació *Whois*, es realitza la consulta d'aquesta al servidor *Whois*.
- Es mostra per pantalla la informació *Whois* obtinguda del servidor.



Il·lustració 17. Diagrama d'estats

4.3. Diagrama de seqüència

Les accions que realitzarà l'aplicació per a complir amb els seus objectius es recullen en el diagrama de seqüència següent:



Il·lustració 18. Diagrama de seqüència

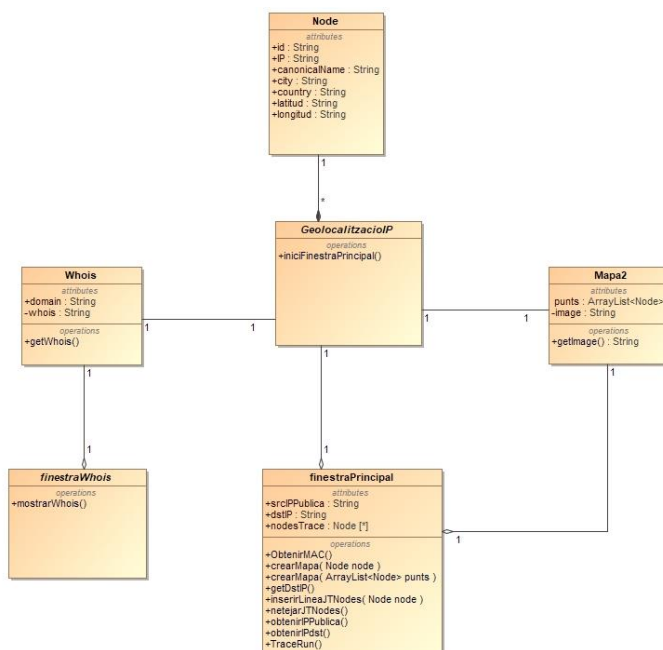
Les accions es detallen a continuació:

1. L'usuari introdueix l'adreça del domini del qual es vol saber els detalls de la traça.
2. L'aplicació comprova que l'adreça introduïda és una adreça vàlida.
3. S'inicia un bucle de cerca de nodes.
4. Es realitza una consulta ICMP modificant el camp TTL.
5. Una vegada es té la resposta del node corresponent, es fa una consulta al servei de geolocalització de les dades geogràfiques del node concret.
6. Es reben al sistema les dades de la consulta.

7. S'afegeixen les dades rebudes en una taula per a la seva posterior utilització.
8. Una vegada finalitzat el bucle de cerca, es mostren les dades per pantalla.
9. Es mostra a l'usuari les dades dels nodes trobats.
10. Igualment es demana al servei de mapes que generi la imatge del mapa amb la informació recollida a la taula de nodes.
11. El servei de mapes retorna al sistema la imatge de mapa generada.
12. El sistema per la seva banda treu per pantalla el mapa creat.
13. L'usuari pot veure per pantalla les dades obtingudes al mapa.
14. L'usuari demana la informació que pot tenir el servei *Whois* del domini demanat.
15. Es passa la sol·licitud feta al sistema per tal que iniciï la consulta.
16. S'identifica a quin continent pertany el domini per fer la consulta al servei *Whois* concret.
17. Es fa la consulta al servei *Whois* concret.
18. El servei *Whois* retorna la informació del domini al sistema.
19. El sistema mostra la informació per pantalla.
20. L'usuari pot consultar la informació proporcionada pel servidor *Whois* per pantalla.

4.4. Diagrama de classes

Per concretar les especificacions dissenyades s'utilitzarà la programació orientada a objectes (POO), amb la qual es defineixen els objectes que seran la base dels processos. En aquesta aplicació els objectes principals són els nodes (*Node*), el mapa (*Mapa2*), i la informació del domini en *Whois* (*Whois*). Amb aquests s'hauran d'implementar els mètodes necessaris per aconseguir els objectius i alguna classe auxiliar per tal d'aconseguir l'aplicació amb finestres que s'ha marcat com objectiu, seguidament es mostra el diagrama de classes creat:



Il·lustració 19. Diagrama de classes

La classe *Node*, contindrà els atributs per tal de definir el node i que després seran necessaris per a la seva manipulació, igualment també s'han de crear els mètodes *setter* i *getter* necessaris:

- *id*: el qual és un número enter que s'utilitzarà per a la seva identificació i se li assignarà per ordre de creació proporcionant informació de la seva posició en la traça.
- *IP*: adreça IP associada al node creat.
- *canonicalName*: aquesta informació es refereix al nom que té assignat el node al servei DNS.
- *city*: ciutat on es troba ubicat el node.
- *country*: país on es troba ubicat el node.
- *latitud*: valor de la latitud geogràfica relacionada amb el node.
- *longitud*: valor de la longitud geogràfica relacionada amb el node.

La classe *Mapa2*, per la seva banda, recull les dades necessàries per crear el mapa i mostrar-lo i conté els atributs:

- *punts*: que és un *ArrayList* amb els nodes creats per la traça.
- *image*: es tracta d'una cadena de text que conte la informació de la pàgina web que mostra el mapa generat.

A més en aquesta classe *Mapa2* s'ha afegit un mètode *getter* per poder accedir al atribut *image*.

L'altre objecte creat és *Whois*, aquest conté els mètodes per obtenir la informació del servidor *Whois* i els atributs per emmagatzemar-la:

- *domain*: paràmetre que rep per tal de saber de quin domini ha de fer la consulta al servei *Whois*.
- *whois*: on s'emmagatzema la informació obtinguda per mostrar per pantalla.

De la mateixa manera s'ha de crear un mètode *getter* per accedir a la informació continguda a l'atribut *whois*.

Segons les indicacions per tal de crear una aplicació de finestres, es recomana que la classe principal no sigui la classe de la finestra principal per tal d'evitar problemes. Per aquest motiu es crearà una classe anomenada *GeolocalitzacioIP*, la qual és l'encarregada de inicialitzar l'aplicació. Aquesta no té cap mètode ni atribut, ja que la seva única funció és inicialitzar la finestra principal.

La classe *FinestraPrincipal*, és la que conté les funcions principals de l'aplicació i des d'on es llancen les altres classes. Així doncs, aquesta classe conté els següents atributs:

- *srcIPPublica*: conté l'adreça IP pública de l'equip on s'executa l'aplicació.
- *dstIP*: s'emmagatzema l'adreça IP del domini objectiu de la traça.
- *nodesTrace*: és una *ArrayList* amb tots els nodes que es trobaran al fer la traça.

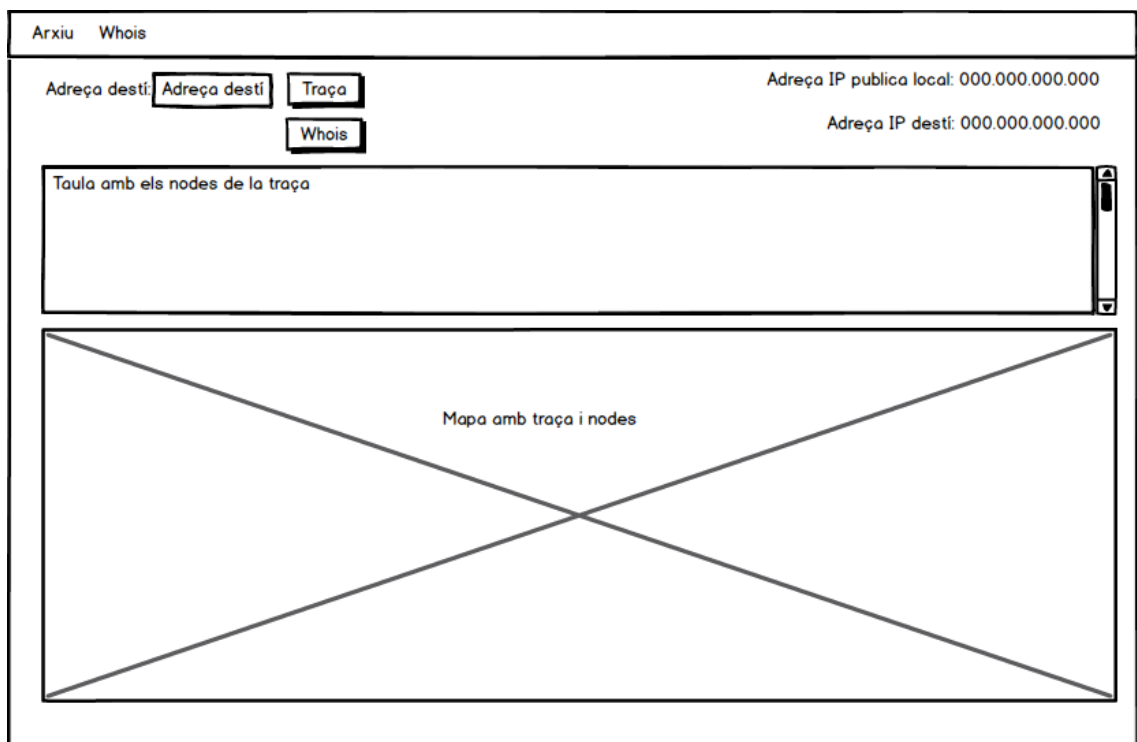
Els mètodes d'aquesta classe *FinestraPrincipal* seran els necessaris per poder portar a terme les funcionalitats demanades:

- *ObtenirMAC*: és l'encarregat de retornar l'adreça MAC de la adreça IP demanada.
- *crearMapa(Node node)*: aquesta funció crearà un mapa amb un node com a centre.
- *crearMapa(ArrayList <Node> punts)*: crea un mapa amb els nodes continguts a la *ArrayList* punts.
- *getDstIP*: mètode *getter* per tal d'accedir a l'adreça IP guardada a *dstIP*.
- *inserirLineaJTNodes*: és el mètode de inserir els nodes a la taula que es mostra a la finestra principal.
- *netejarJTNodes*: esborra els nodes que es mostren a la taula de la finestra principal.
- *obtenirIPPublica*: retorna l'adreça IP pública de l'equip en el que s'està executant l'aplicació.
- *obtenirIPdst*: retorna l'adreça IP del domini sol·licitat.
- *traceRun*: mètode que realitza la consulta ICMP i emmagatzema per a cada resposta el node que la *ArrayList nodesTrace*.

Finalment tenim la classe *FinestraWhois* que, com a únic objectiu, ha de mostrar les dades obtingudes a *Whois* en una finestra a part de la finestra principal.

4.5. Disseny de les finestres

El disseny que es vol crear és d'aspecte senzill, sense gaires elements a la finestra per tal de no carregar-la visualment i a la vegada es vol que aquesta mostri tota la informació demanada. Per aquest motiu es vol crear una finestra principal com la que es mostra a continuació.



Il·lustració 20. Disseny finestra principal

En la qual hi haurà una espai per a ficar el domini o adreça IP que es vol utilitzar, al costat es mostraran dos botons per a realitzar les dues tasques principals de l'aplicació. A la dreta es mostrarà la informació relativa a l'adreça IP pública de l'equip local i l'adreça IP de destí.

A sota d'aquesta informació es mostrarà una taula amb la informació relativa a cada node, que s'anirà incrementant a cada moment. I a la part inferior es mostrarà el mapa generat.

5. Implementació del producte

A l'hora de crear el producte, s'han de fer els paquets ICMP que s'enviaran a través de la xarxa per a realitzar el procés de traça que ens identificarà els diferents nodes entre l'origen i el host de destí. Aprofitant la llibreria *Jpcap*, s'han implementat els paquets tant a nivell de xarxa o *Ethernet* com a nivell IP, tal com s'explicarà més endavant.

Per poder crear els paquets en aquests nivells s'ha d'obtenir les següents dades:

- Adreça IP de destí.
- Adreça IP pública d'origen.
- Adreça MAC d'origen.
- Adreça MAC de la porta d'enllaç, que és el destí del primer salt que farà el paquet, i qui després s'encarregarà de redirigir els paquets al següent destí a nivell de xarxa.

Una vegada creats els paquets amb aquestes dades es realitzarà un bucle incrementant el valor *TTL* en cada paquet i s'enviarà, i aprofitant les funcionalitats de la llibreria *Jpcap* es capturaran les respostes i es processarà la informació rebuda.

El següent pas, tal com s'ha dit, és processar la informació, és a dir, amb la adreça IP de resposta s'utilitzarà un servei de geolocalització, *ip-api.com*, per tal de saber la latitud, longitud, ciutat, i país d'aquesta adreça. També s'utilitzarà una altra llibreria de Java per a consultar el *canonical name* de cada node pel qual es coneix aquest als servidors DNS, i que s'anomena *InetAddress*.

A continuació, amb tota aquesta informació s'implementarà el mapa amb la localització dels nodes i la ruta que segueixen els paquets durant el procés de comunicació. Aquesta funcionalitat es crea utilitzant *Google Maps Script*, implementant una pàgina web local que realitza la consulta a aquesta API i mostrant-la en l'aplicació gracies a les llibreries de *JavaFX*, que ens permeten mostrar pàgines web en un *JFXPanel*.

Altra funcionalitat que s'inclou en l'aplicació és la possibilitat de consultar la informació que hi ha als servidors de el servei *Whois* sobre el domini de destí. En aquest cas la consulta es fa directament al servei *Whois*, però cal tenir en compte que aquest servei divideix els servidors per continents i cal consultar al servidor correcte per obtenir la informació detallada. Per a realitzar aquesta tasca, primer es fa una consulta a un arxiu en formant CSV, per a identificar a quin continent pertany l'adreça i quin és el servidor d'aquest continent, després només es necessari fer la consulta a aquest servidor en concret.

Finalment, cal implementar l'aplicació dintre d'un entorn de finestres, tal com fixa l'objectiu, i per a complir amb aquest utilitzarem les llibreries de *Java Swing*, les quals ens permeten crear finestres amb botons i menús tal com s'ha dissenyat.

A continuació es detallarà com s'ha realitzat cada pas.

5.1. Creació de paquets i traça

Tal com ja s'ha dit, per poder realitzar la traça dels paquets s'utilitzaran les funcionalitats de la llibreria *Jpcap*. Aquesta llibreria té com a objectiu realitzar tasques de captura i anàlisi

dels paquets de xarxa que es reben per un adaptador de xarxa concret. Encara que el nostre objectiu no es crear una aplicació *sniffer*, si que es poden aprofitar aquestes característiques per a implementar l'aplicació, ja que ens permet enviar i analitzar paquets de xarxa. El funcionament d'aquesta és el següent:

- Identificar amb quin adaptador de xarxa es treballarà, per tal que aquesta analitzi els paquets que entren per aquest.
- Inicialitzar el captador de paquets de *Jpcap* indicant l'identificador de l'adaptador de xarxa.
- Crear i enviar els paquets ICMP.
- Crear un bucle que analitza el tràfic que es rep per l'adaptador de xarxa i només capturar els paquets del tipus que s'ha especificat, en aquest cas les respostes ICMP. Aquest bucle finalitzarà en el moment de que s'arribi al host destí o es superi en nombre màxim de salts.

Així doncs, s'ha creat el codi necessari per a poder identificar l'adaptador de xarxa amb el que l'aplicació treballarà, el qual primer crea una llista d'adaptadors que té la màquina origen i després utilitzant l'adreça MAC de l'adaptador local es realitza una cerca per aquesta llista fins trobar l'adreça MAC que coincideix, obtenint així l'identificador que s'utilitzarà per inicialitzar el captador.

S'ha inclòs una funció denominada *ObtenirMAC()* que retorna l'adreça MAC de l'equip i permet així buscar-la en el llistat que retorna *JpcapCaptor.getDeviceList()*.

En aquest moment ja es pot inicialitzar el captador de paquets amb l'identificador de l'adaptador de xarxa, tal com es mostra en el codi següent:

```
//Inicialitzem captador de paquets.  
NetworkInterface device=list[srcId];  
captor=JpcapCaptor.openDevice(device,2000,false,1000);
```

Una vegada inicialitzat el captador de la llibreria *Jpcap*, és el moment de preparar els paquets ICMP que s'enviaran a través de la xarxa cap al equip destí. Durant el procés d'inicialització del captador ja hem obtingut l'adreça MAC de l'equip d'origen, igualment tenim l'adreça IP de destí, ja que per començar el procés de traça cal introduir el domini o l'adreça IP de destí (tant si s'introdueix una o altra el programa obté o verifica l'adreça IP de destí), i també es té l'adreça IP d'origen la qual s'obté automàticament al iniciar el programa. Així doncs, cal trobar l'adreça MAC del destí del primer salt que farà el paquet, i aquesta és la de la porta d'enllaç o *gateway*. Per obtenir aquesta adreça existeixen diferents maneres, però en aquesta aplicació s'ha decidit aprofitar que es té inicialitzat el captador de paquets de *Jpcap* i enviar una consulta a una pàgina web que es sap que contestarà com és *google.com*. Es

configura el captador de paquets per tal que només escolti els paquets TCP a la adreça IP de l'equip d'origen. Una vegada es rep la resposta només cal extreure el valor de l'adreça MAC de destí del paquet que s'envia.

En aquest moment, ja es poden generar els paquets de consulta ICMP que s'enviaran a través de la xarxa per a identificar els equips intermedis. Per aquesta tasca *Jpcap* proporciona dos mètodes, un per a crear els paquets ICMP i l'altre per a generar el paquet de xarxa. Aquests faciliten molt la tasca de crear els paquets donat que només cal proporcionar les dades ja obtingudes.

El camp de dades es de 32 bytes i s'ha emplenat amb 0 i el valor del camp verificació (*checksum*) també s'ha establert a 0.

L'últim pas abans de iniciar la captura de la traça és inicialitzar el filtre del captador per tal que aquest pugui analitzar les dades necessàries retornades pels equips intermedis. Això es realitza amb la comanda que es mostra a continuació:

```
//Es genera el filtre per capturar nomes els paquets icmp amb destí host.  
captor.setFilter("icmp and dst host "+srcIP.getHostAddress(), true);
```

Finalment, es crea el bucle per tal de capturar tots els paquets ICMP rebuts i analitzar la resposta segons el codi ICMP, així mateix dintre d'aquest bucle es fa una crida a la classe *Node* per a generar, per a cada equip intermedi, un objecte d'aquesta classe amb les seves dades i s'emmagatzema aquest en una taula que servirà, al finalitzar el procés de traça, per generar el mapa. Un dels paràmetres inicialitzats abans del bucle és l'objecte *JpcapSender* el qual és l'encarregat d'enviar el paquet ICMP, a més s'ha establert un límit de salts de 30 que és el nombre de salts màxim habitual per tal d'arribar d'un equip a un altre a Internet. Altre paràmetre important és *hop_limit* que és el valor de TTL i que s'incrementarà cada cop al bucle.

5.2. Geolocalització IP dels nodes.

Durant el procés de detecció dels nodes, cada cop que es rep la notificació d'un salt es genera un objecte *Node*, el qual conté tota la informació necessària d'aquest (identificador, adreça IP, *canonical name*, país, ciutat, latitud i longitud). Per a generar aquests objectes *Node* s'ha creat un mètode constructor el qual rep el identificador (que serveix per a situar-lo en la posició correcta de la taula) i l'adreça IP, amb aquestes dades es genera l'objecte.

A l'hora de crear un objecte de la classe *Node* s'han tingut en compte dos possibilitats:

- Es tracta d'una resposta de temps excedit, la qual cosa significa que el salt corresponent ha trigat massa temps en respondre. Llavors l'adreça IP es considera com *Timeout* i els camps restants s'emplenen amb un guio ("") per expressar que estan buits.
- Es tracta d'una resposta amb una adreça IP vàlida, en aquest cas es realitza la consulta de les dades al servei *ip-api.com*. Aquest servei retorna les dades demanades mitjançant un número (que es proporciona a la pàgina web del servei) en format CSV (*comma-separated values*). Una vegada rebudes les dades a través de la consulta a la

URL del servei, aquestes es llegeixen i es copien en un taula per poder assignar els valors als paràmetres corresponents de l'objecte.

També hi ha dues possibilitats quan el node té una adreça IP, és a dir, respon a la consulta ICMP, aquestes són:

- L'adreça IP pertany al rang d'adreces reservades per a xarxes privades.
- L'adreça IP pertany al rang d'adreces públiques.

En el primer cas, *Ip-api.com* retorna un missatge indicant que es tracta d'una adreça IP privada i el primer camp conté el valor *fail* i el segon camp el valor *private range*, aquest segon camp s'assigna al paràmetre de l'objecte *canonical name* per tal d'indicar que es tracta d'una adreça IP privada.

En el segon cas, es capturen els valors necessaris per a identificar el node i s'assignen als paràmetres corresponents.

5.3. Mostrar informació en un mapa

L'objectiu de mostrar la informació dels nodes en un mapa és veure de manera més clara el recorregut que realitzen els paquets a través d'Internet fins arribar al destí, indicant la ubicació geogràfica d'aquests. Tal com s'ha descrit en l'apartat d'anàlisi de recursos, la opció triada per a complir aquest objectiu ha estat *Google Maps JavaScript*. El funcionament d'aquest servei es basa en fer una consulta *JavaScript* a l'API de *Google*, aquesta consulta ens mostra a través de una pàgina web el mapa amb les dades demanades. Així doncs, cal tenir en compte dos tasques principals per tal d'integrar el mapa en l'aplicació:

- Generar una pàgina web, crear la consulta *JavaScript* dintre d'aquesta, i rebre el mapa generat.
- Inserir a l'entorn gràfic un panell on es pugui visualitzar aquesta pàgina web.

La primera tasca es realitza mitjançant la classe *Mapa2*, aquesta rep els nodes que ha de recollir el mapa i s'encarrega de generar la pàgina web (arxiu en format *html*) amb la consulta *JavaScript*.

En primer lloc es genera l'arxiu *html* en la mateixa ubicació on es troba l'aplicació, i es guarda com a paràmetre la ruta d'aquest, tal com es mostra en el codi següent:

```
File f=new File("mapa.html");
this.image=f.getAbsolutePath();
FileWriter mapa = null;
```

A continuació, s'extreu les dades de latitud i longitud de cada node mitjançant un bucle i es va emplenant una cadena de text que serà posteriorment guardada al fitxer *mapa.html* creat a l'inici. Amb aquestes dades se li proporciona a l'API de *Google* la posició de cada node i es genera la ruta denominada *polilinea* i els marcadors de cada node identificat denominats *markers*. Igualment important és inserir la clau proporcionada per *Google* per a identificar el

projecte, *key*. També cal tenir en compte que no s'inclouen els nodes amb IP privada ni els nodes que no han contestat (*timeout*) al no tenir dades de latitud i longitud.

En aquesta classe *Mapa2*, s'ha afegit un altre mètode constructor amb les mateixes característiques que el que s'ha descrit però en aquest cas només es rep el paràmetre del node inicial que és on es troba el equip origen. La funció d'aquest és crear el mapa que es mostra al inicialitzar la aplicació.

Finalment, per poder mostrar el mapa a l'aplicació es crea un panel on es mostri la pàgina web generada amb el mapa, per a fer-ho possible s'utilitzen diferents llibreries de *JavaFX* les quals permeten crear un visor web que es pot cridar des de l'aplicació.

5.4. Informació Whois

Aquesta informació s'ha decidit mostrar-la en una finestra diferent a la principal, i que pot aparèixer en qualsevol moment del procés de traça si es demana. Com s'ha descrit, aquesta funció té per objectiu mostrar la informació disponible al servidor corresponent al continent de l'adreça destí. Aquesta informació estarà continguda en un objecte anomenat *Whois*, el qual contindrà una cadena en format *html* amb la informació rebuda del servidor.

Així doncs, en primer lloc es busca el continent, i per tant el servidor de dades de *Whois*, al que pertany el equip de destí de la consulta. Això es realitza aïllant el primer número decimal de l'adreça IP destí, i després buscant el servidor que li correspon en una taula en format CSV.

Una vegada es té el servidor concret, cal obrir un *socket* de comunicació amb aquest servidor amb el port de destí 43, que s'utilitza per a fer les consultes. El següent pas, una vegada obert el *socket* de comunicació, és enviar l'adreça IP del domini que es vol consultar i es rep tota la informació referent a aquest node que hi ha en el servidor. I tal com ja s'ha dit, es guarda en una cadena de text en format *html*. A continuació es pot veure una imatge de com es mostra la informació en la finestra.



```
WHOIS
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/whois_tou.html
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/whois_reporting/index.html
#
#
# Query terms are ambiguous. The query is assumed to be:
# "n 172.217.16.238"
#
# Use "?" to get help.
#
NetRange: 172.217.0.0 - 172.217.255.255
CIDR: 172.217.0.0/16
NetName: GOOGLE
NetHandle: NET-172-217-0-1
Parent: NET172 (NET-172-0-0-0)
NetType: Direct Allocation
OriginAS: AS15169
Organization: Google LLC (GOGL)
RegDate: 2012-04-16
Updated: 2012-04-16
Ref: https://whois.arin.net/rest/net/NET-172-217-0-1

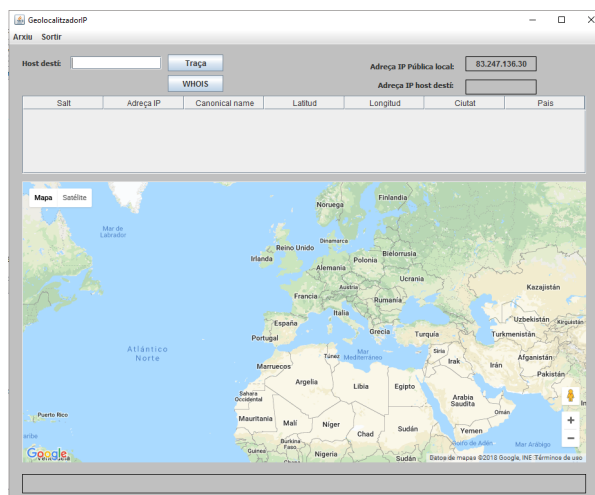
OrgName: Google LLC
OrgId: GOGL
Address: 1600 Amphitheatre Parkway
City: Mountain View
StateProv: CA
PostalCode: 94043
Country: US
RegDate: 2000-03-30
Updated: 2017-12-21
Ref: https://whois.arin.net/rest/org/GOGL
```

Il·lustració 21. Finestra Whois

5.5. Entorn gràfic de l'aplicació

En aquest punt també s'ha seguit el disseny realitzat en la fase de disseny, així doncs l'aplicació consta d'una finestra on es distribueix la informació que s'obté de forma que sigui fàcil visualitzar-la per part de l'usuari.

Inicialment, apareix a la finestra la taula de nodes buida i el mapa sense cap punt definit encara, també es pot apreciar que hi ha un espai en blanc on s'ha d'introduir el domini o l'adreça IP d'aquest, i al costat dos botons per a realitzar les accions de *Traça* i *Whois*. A la part superior dreta també s'ha inclòs la informació de l'adreça IP pública local i quan s'introdueix el domini o l'adreça IP de destí, també apareix l'adreça IP de destí. Altre component que s'ha afegit, seguint el disseny és un menú a la barra superior de l'aplicació on es pot triar entre Arxiu (amb la possibilitat de guardar o obrir una traça) o sortir. En aquest punt, si que s'ha diferit del que s'havia dissenyat inicialment, la raó és que s'ha vist innecessari afegir dos camins per a fer una mateixa funció, així doncs no s'ha ficat l'opció *Whois* a la barra i s'ha substituït per l'opció de sortir, per facilitar la possibilitat de tancar l'aplicació des d'aquesta. Finalment, també s'ha afegit un requadre a la part inferior per tal de delimitar l'espai per on apareixeran els missatges d'error, si n'hi ha, com poden ser que la URL ha estat mal introduïda o si s'han excedit el nombre màxim de salts.



Il·lustració 22. Finestra principal (Inici)

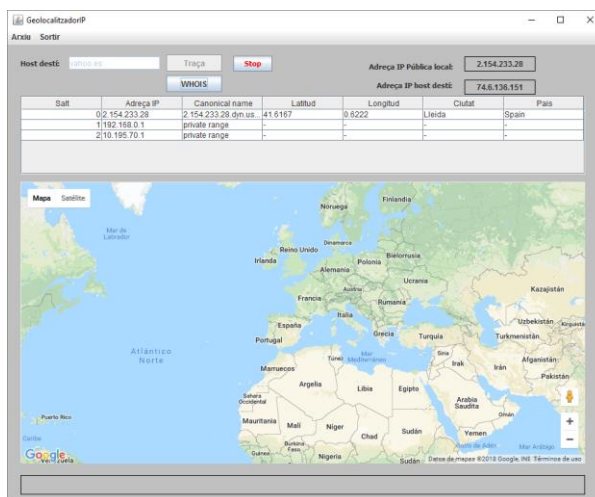
La implementació de l'entorn gràfic s'ha realitzat amb les llibreries de *Java Swing*, aprofitant les possibilitats que facilita el IDE *Oracle Netbeans*. Des d'aquest es pot accedir a la finestra de disseny de la classe del tipus *JFrame* i permet afegir els objectes necessaris a aquesta (botons, menús, taules, *scrolls*,...) i programar el seu comportament d'una manera molt intuïtiva.

Altre punt important de la implementació de l'entorn gràfic és el comportament dels diferents components durant l'execució de l'aplicació. Sobre aquest aspecte, s'han pres les següents decisions:

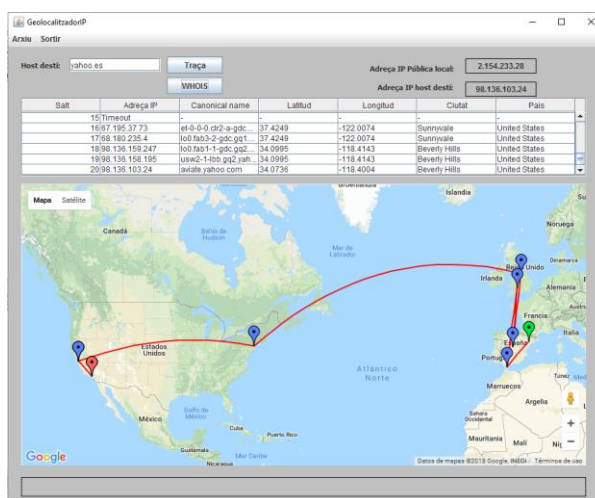
- A l'inici tots els components estan actius, si es prem el botó traça sense introduir cap adreça, apareix un missatge d'error. Si es prem el botó *Whois*, aquest no mostra res.

- Quan s'inicia el procés de traça, el botó de traça es desactiva per evitar que es torni a prémer i llançar més d'una traça alhora. Per tal d'aturar la traça abans d'acabar, apareix un botó *Stop* al costat del botó desactivat de *Traça* que permet fer aquesta acció. Igualment, apareix l'adreça IP de destí al quadre de text destinat per això. A la taula de nodes es van afegint els salts realitzats amb les dades corresponents, en el moment que la taula està plena apareix una barra *scroll* al costat de la taula que es situa a la part inferior per tal de poder veure el nodes que es van afegint, també aquest *scroll* permet desplaçar la taula per revisar tots els nodes.
- A la finalització de la traça, desapareix el botó *Stop* i torna a quedar actiu. A la vegada, es mostra el mapa amb els nodes identificats i la traça trobada.
- Si apareix un missatge d'error, es queda tot com a l'inici, quan es comença una nova traça correcta, aquest missatge desapareix.

Seguidament es mostren les pantalles amb les diferents opcions:



Il·lustració 23. Finestra principal durant el procés de traça



Il·lustració 24. Pantalla principal al finalitzar el procés de traça

Per habilitar i deshabilitar els botons s'ha configurat a les accions dels botons corresponents el valor de les opcions *setvisible* i *setenable* com *true* o *false* segons correspongui.

També s'ha inclòs una barra de menú on es pot trobar dos opcions, *Arxiu* i *Sortir*. Dintre del primer menú, *Arxiu*, s'ha implementat l'opció de guardar les dades de la traça en un arxiu en format CSV que es pot emmagatzemar en l'equip origen per tal de recuperar quan es desitgi. Igualment, s'ha inclòs l'opció de recuperar aquest arxiu CSV i visualitzar la traça a la taula i el mapa. Aquestes dues opcions s'han creat utilitzant les funcionalitats de la llibreria *JFileChooser*, que permet obrir una finestra per explorar les carpetes del sistema i introduir el nom del fitxer que es vol guardar o recuperar el nom d'aquest fitxer. Una vegada creat o seleccionat el fitxer, aquesta retorna la ruta del mateix per tal de poder escriure'l o llegir-lo.

L'altra opció només té la funció de facilitar el tancament de l'aplicació quan l'usuari ho desitgi mitjançant la instrucció *System.exit(0)*.

6. Conclusions

Un cop finalitzada la implementació del producte és moment d'avaluar si s'han assolit els objectius inicials que es plantejaven i crec que es pot afirmar que, certament aquests objectius s'han assolit. S'ha implementat una aplicació gràfica que identifica els nodes que hi ha entre un equip d'origen i un de destí durant un procés de comunicació, aquests nodes es poden veure dintre d'un mapa juntament amb el camí que recorren els paquets, i a més es mostra la informació concreta dels nodes identificats, així com la informació que hi ha al servei *Whois* del node destí. I tot això s'ha fet implementant els paquets ICMP per tal de poder fer les consultes.

Al començament d'aquest treball, el problema més gran que es plantejava era com poder crear les consultes ICMP i capturar la resposta sense utilitzar programes ja creats com *Ping* o *Traceroute*, però durant el desenvolupament del projecte he pogut aprendre com aquest procés es pot realitzar utilitzant llibreries de *Java* i, a més, la utilització d'aquestes, dona major llibertat a l'hora de programar. El no utilitzar un programa del sistema operatiu concret, permet que l'aplicació sigui més exportable a altres sistemes operatius, com Linux (encara que no s'ha provat l'aplicació en aquest).

Altre objectiu assolit, en aquest cas personal, és aprendre a treballar amb serveis disponibles a la xarxa. Ha estat una experiència molt enriquidora en aquest aspecte, ja que, el procés d'avaluació dels diferents serveis de la xarxa i la posterior elecció i implementació m'ha donat l'oportunitat de desenvolupar coneixements sobre com es poden utilitzar aquests recursos.

En el aspecte de la localització d'equips a través de la seva adreça IP, s'ha comprovat com la fiabilitat de les dades dels nodes és molt relativa, depèn molt del servei que es consulta i si aquest està actualitzat. Normalment els serveis més actualitzats solen ser els de pagament, això pot ser per que hi ha darrera una feina d'actualització més constant. També s'ha vist com finalment són els ISP qui assigna les adreces IP als equips, i d'aquesta manera són ells qui saben concretament a quina zona geogràfica concreta correspon una adreça, és per això que resulta molt difícil ubicar una adreça IP a una latitud i longitud exacta si el ISP no ha facilitat aquesta informació al servei concret.

Altra conclusió que s'ha extret una vegada comprovat el funcionament de l'aplicació és que els paquets realitzen molts salts entre llargues distàncies per arribar al seu destí, això s'ha pogut comprovar fent la traça de dominis molt coneguts com *google.es*, que encara que només s'identifiquen alguns salts, hi ha molts més que no responen a la consulta ICMP i que també hi són.

En l'apartat referent al seguiment de la planificació, aquesta ha estat molt adequada, ja que s'ha pogut seguir el ritme marcat sense realitzar correccions. Fins i tot, en algun moment s'ha treballat avançant feina de les tasques següents. Així mateix, la metodologia aplicada, en tot moment ha facilitat avançar d'una manera ordenada i incremental.

Finalment, cal expressar que la realització d'aquest TFG ha estat per a mi un procés molt positiu al poder crear un producte d'una manera global i totalment funcional.

7. Glossari

Canonical name: és un tipus de registre que es troba als servidors DNS i serveix per a indicar un al·lies de un nom de domini.

DDoS: (*Distributed Denial of Service*) és un tipus d'atac a través de la xarxa que consisteix en generar moltes consultes a un equip objectiu des d'un gran nombre d'equips atacants, de manera que s'aconsegueix que l'equip objectiu no pugui atendre les demandes i es bloquegi.

DNS: (*Domain Name System*) conjunt de servidors que emmagatzemen els noms dels dominis relacionats amb la seva adreça IP, la seva funció és resoldre les consultes dels dominis retornant la adreça IP corresponent.

Gateway: o porta d'enllaç, és l'equip de la xarxa encarregat de fer l'encaminament dels paquets cap a fora d'una xarxa o redirigir-los als altres equips de dintre de la xarxa local.

Geocodificació: terme que es refereix a assignar referències geogràfiques de longitud i latitud a punts concrets d'un mapa.

Geolocalització: capacitat per obtenir la ubicació geogràfica real d'un objecte com pot ser un radar, un telèfon mòbil, o un ordinador connectat a Internet.

Host: terme que s'utilitza a la informàtica per a referir-se als equips que estan connectats en una xarxa.

ICMP: protocol de missatges de control d'Internet, és un protocol que facilita el diagnòstic de possibles fallades de la xarxa mitjançant l'enviament i recepció de paquets específics.

JavaScript: llenguatge de programació orientat a objectes, molt utilitzat en la programació web que permet interfícies d'usuari i pàgines web dinàmiques.

Jpcap: llibreria específica de Java que permet treballar amb paquets de comunicació i realitzar tasques de captura de comunicacions, pot treballar amb LibPcap i WinPcap.

Ping: aplicació específica de Microsoft Windows que permet enviar consultes ICMP per a realitzar tasques de diagnòstic de la xarxa.

Sniffer: es refereix a un tipus d'aplicació que s'utilitza per a realitzar captura de paquets de la xarxa per tal de poder analitzar-los i realitzar anàlisis de la xarxa.

Traceroute: aplicació específica que s'utilitza per a realitzar la traça des d'un equip local que l'executa a un altre equip remot, permet visualitzar els equips de comunicació que hi ha entre aquests.

Whois: servei que permet fer consultes de qui és el propietari d'un determinat domini.

WinPcap: aplicació que treballa en un sistema operatiu Windows i que permet accedir a les comunicacions a nivell de paquets i datagrames.

8. Bibliografia

1. IANA. *IANA OPv4 Address Space Registry*. <https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>
2. IANA. *Internet Control Message Protocol ICMP*. <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml>.
3. IANA. *Numbers Resources*. <https://www.iana.org/numbers>.
4. IETF. (1981). *RFC792- Internet Control Message Protocol*. <https://tools.ietf.org/html/rfc792>
5. *Protocolo ICMP. Protocolo de Mensajes de Control de Internet*. En *PC Paso a Paso* num. 23. Pag 14-37. https://www.nebrija.es/~cmalagon/seguridad_informatica/Lecturas/TCP-V_ICMP_hxc.pdf
6. Cisco Systems. (2006). *Understanding the Ping and Traceroute Commands*. <https://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html>
7. IETF. (1989). *RFC1122-Requirements for internet hosts- Communication Layers*. <https://tools.ietf.org/html/rfc1122#page-38>
8. ICANN. *ICANN WHOIS*. <https://whois.icann.org/es>.
9. DAVIS, CHRISTOPHER. (2001). *DNS LOC Geo-enabling the Domain Name System*. <https://www.ckdhr.com/dns-loc/>
10. CANTERO, GUSTAVO. (2009). *Obtener la Ciudad i el país de una dirección IP en Programando a Medianoche*, 24 de maig. <http://www.programandoamedianoche.com/2009/05/obtener-la-ciudad-y-el-pais-de-una-direccion-ip/>
11. *Qué es la geolocalización por IP y cuanto revela tu dirección física en Xataka*, 5 de juny de 2017. <https://www.xataka.com/basics/que-es-y-cuanto-revela-de-tu-la-geolocalizacion-por-ip>
12. ARDIÓNS, ANDREA. (2017) *¿Quieres localitzar direcciones IP? Descubre 3 servicios web gratis que funcionan en ProfessionalReview*, 7 de febrer. <https://www.profesionalreview.com/2017/02/07/localizar-direccion-ip-servicios/>
13. AWAIS, AHMAD. (2017). *What is the best API for geolocating an IP address?* en QUORA. 20 d'abril. <https://www.quora.com/What-is-the-best-API-for-geolocating-an-IP-address>
14. AROCHE, JAVIER.(2010). *Como saber la ubicación de un usuario en Maestros del Web*, 4 de febrer. <http://www.maestrosdelweb.com/detectar-ubicacion-usuario-geolocalizacion/>
15. KUROSE, JAMES F. i ROSS, KEITH W. (2013). *Computer Networking. A top-down approach. Sixth Edition*. England: Editorial Pearson.
16. RUSTY HAROLD, ELLIOTTE. (2013). *Java Network Programming. Fourth Edition*. Estats Units d'Amèrica: Editorial O'Reilly.
17. WIELENGA, GEERTJAN. (2015). *Beginning Netbeans IDE. For Java Developers*. Estats Units d'Amèrica: Editorial Apress.

9. Annexos.

9.1. Annex A. Manual de instal·lació.

Aquesta aplicació s'ha desenvolupat, tal com ja s'ha exposat durant la memòria, amb el llenguatge de programació Java. Es per aquest motiu que es indispensable per tal que funcioni tenir instal·lada la versió corresponent de Java. A continuació s'enumeren els requisits necessaris per a poder utilitzar l'aplicació:

- Sistema operatiu Windows 64 bits.
- *Java* versió 8 o superior.
- *WinPcap* versió 4.1.3.
- *Jpcap* versió 0.7.

Seguidament es farà una explicació de com cal instal·lar aquest programari per poder utilitzar sense problemes l'aplicació presentada en aquest TFG.

9.1.1. *Java*.

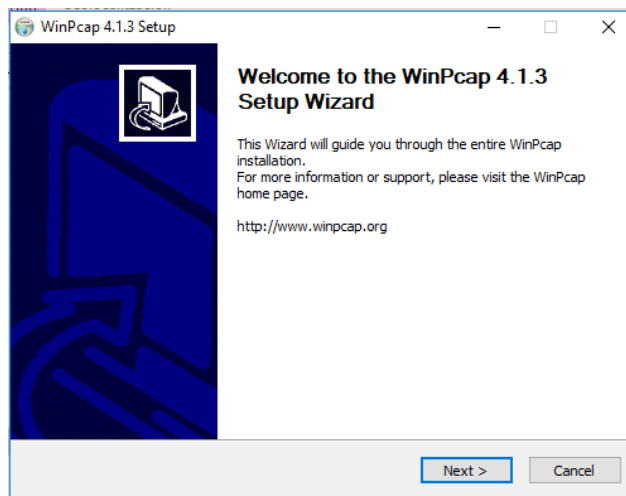
Per tal d'instal·lar la versió correcta de Java per a Windows, només cal accedir a la següent adreça de Internet <https://www.java.com/es/download/win10.jsp>. Una vegada en aquesta, cal descarregar la versió de Java que apareix (que és la última versió disponible), i executar l'arxiu que ens proporcionen. A partir d'aquest punt, només cal acceptar totes les opcions que ens indiquen i realitzar la instal·lació al nostre equip.

Una vegada finalitzada la instal·lació l'aplicació ja s'iniciarà, però la part relativa a la traça de nodes i mostra de mapes, encara no funcionarà, ja que aquesta part utilitza la llibreria *Jpcap* per a crear i capturar els paquets ICMP.

9.1.2. *WinPcap*.

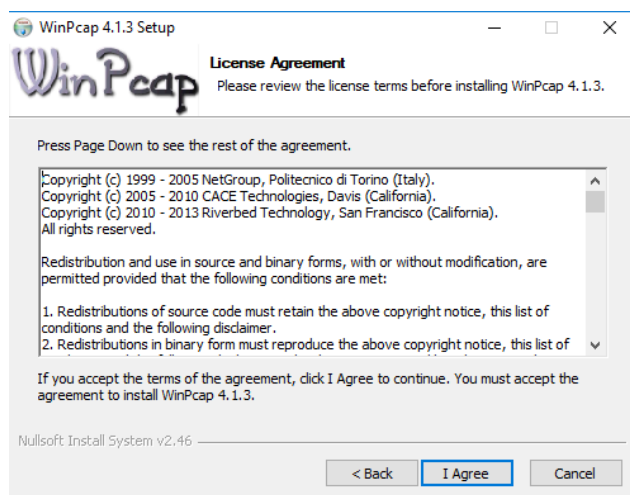
WinPcap es pot descarregar de l'adreça <https://www.winpcap.org/install/>, una vegada descarregat s'han de seguir els següents passos:

1. Executem l'arxiu i apareix una finestra com la que es mostra a continuació on caldrà prémer el botó "Next".



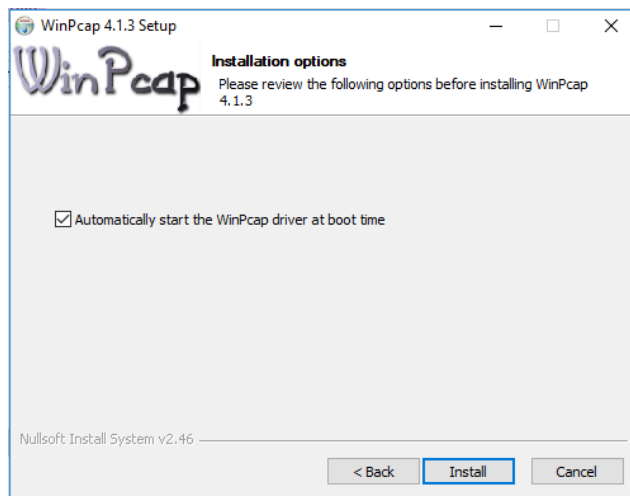
Il·lustració 25. WinPcap, inici d'instal·lació.

2. En la següent finestra es mostra la llicència d'utilització del programa, serà necessari acceptar aquesta per poder prosseguir amb la instal·lació, per fer-ho s'ha de prémer el botó "I Agree" .



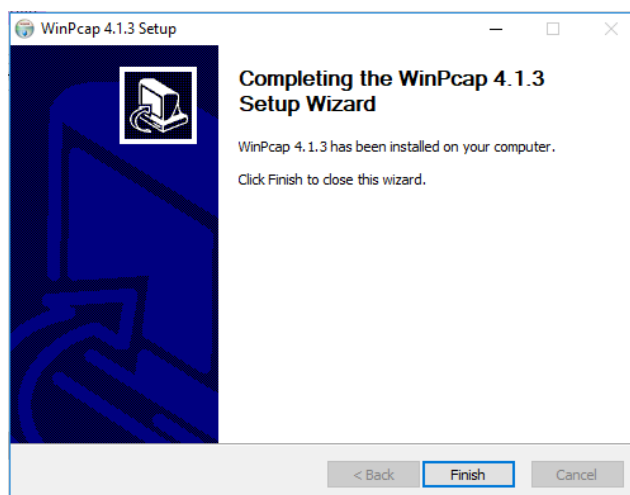
Il·lustració 26. WinPcap, acceptació de llicència.

3. A continuació la instal·lació dona l'opció de configurar que *WinPcap* s'executi automàticament al inici del sistema operatiu, marquem aquesta opció per assegurar-nos que està el servei iniciat quan executem l'aplicació. Una vegada marcada la casella corresponent, premem el botó "Install".



Il·lustració 27. WinPcap, configuració.

4. Finalment, si el procés d'instal·lació s'ha realitzat correctament se'ns mostrarà la corresponent finestra de finalització de la instal·lació i en aquesta polsarem el botó "Finish" per finalitzar el procés.



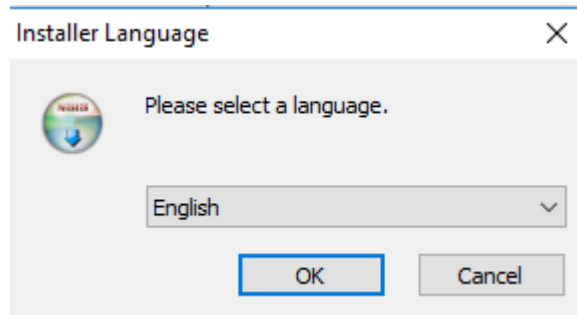
Il·lustració 28. WinPcap, finalització d'instal·lació.

9.1.3. Jpcap.

Una vegada tenim instal·lada la versió correcta de Java i la versió corresponent de *WinPcap*, cal realitzar la instal·lació de la llibreria *Jpcap*. Aquesta instal·lació es realitza en dos fases, una automatitzada que instal·la el client de 32 bits i una altra manual en la qual cal copiar les llibreries *dll* per a Windows 64 bits (que actualment és la versió més utilitzada). Seguidament es detalla aquesta instal·lació:

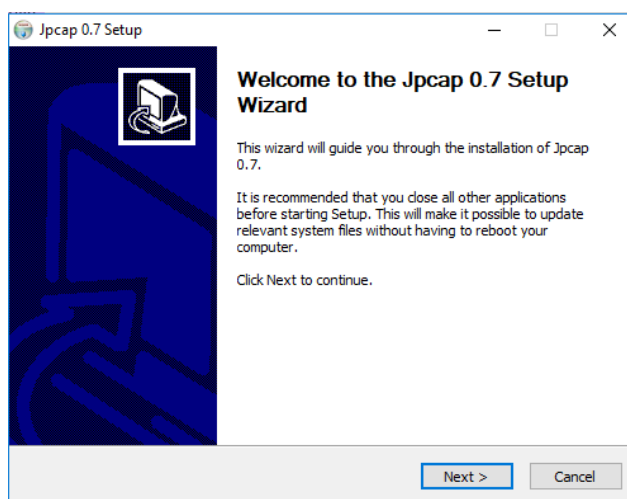
1. La versió 0.7 de *Jpcap* es pot descarregar de la següent adreça de Internet, https://sourceforge.net/projects/ridsystem/files/necessary%20libraries/JpcapSetup-0.7.exe/download?use_mirror=netcologne&download=&failedmirror=jaist.dl.sourceforge.net

ceforge.net , o es pot fer una cerca d'aquesta ja que és una llibreria que es troba amb facilitat a la xarxa. Una vegada descarregada, executem l'arxiu JpcapSetup-0.7 descarregat, i apareix la finestra inicial d'instal·lació següent, on se'ns demana que triem l'idioma de la instal·lació, només deixa seleccionar *English*, i premem el botó "OK".



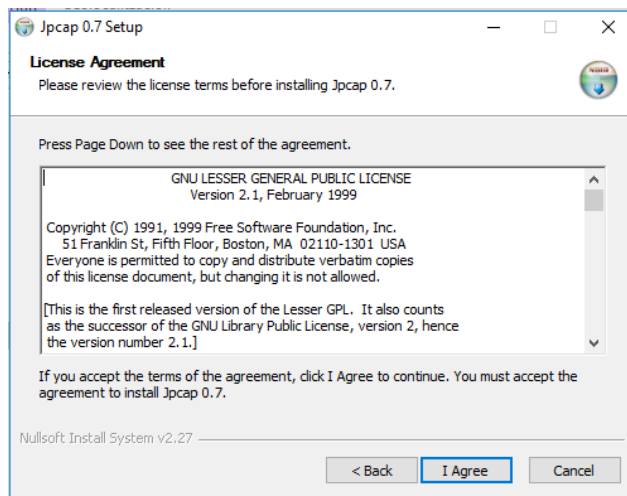
Il·lustració 29. Jpcap, inici instal·lació.

2. La següent finestra que apareix ens dona la benvinguda a la instal·lació i només dona l'opció de cancel·lar o seguir amb aquesta, premem el botó "Next".



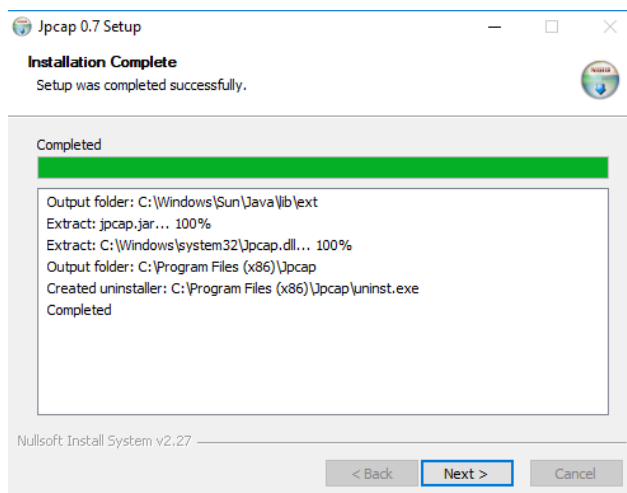
Il·lustració 30. Jpcap, benvinguda instal·lador.

3. En aquest moment es mostra la llicència d'us que igualment haurem d'acceptar per poder continuar amb la instal·lació, premem el botó "I Agree".



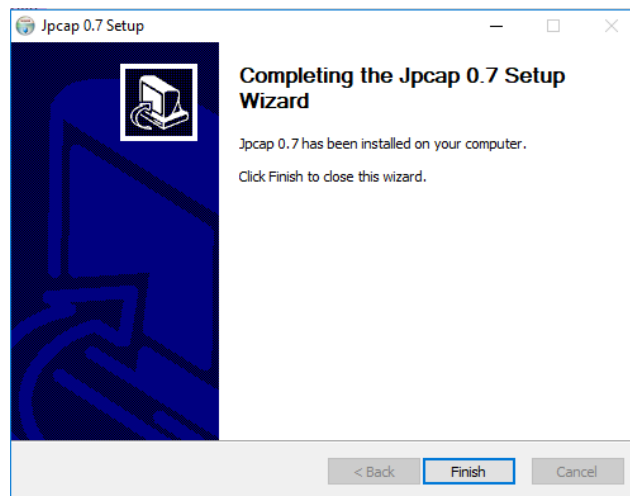
Il·lustració 31. Jpcap, acceptació de llicència.

4. Veurem una pantalla on es mostra el progrés d'instal·lació i una vegada aquest ha finalitzat, haurem de prémer el botó "Next" per avançar a la següent finestra.



Il·lustració 32. Jpcap, procés d'instal·lació.

5. Finalment es mostra una finestra informativa indicant que el procés d'instal·lació ha finalitzat, premem "Finish" per acabar amb aquest i es tanca la finestra.



Il·lustració 33. Jpcap, finalització del procés.

En aquest punt de la instal·lació de *Jpcap* ja tenim la llibreria per a equips amb sistema operatiu Windows amb 32 bits, però cal copiar les llibreries *jpcap.dll* per a Windows 64 bits per que l'aplicació funcioni correctament en aquests sistemes. Això es farà tal com s'explica seguidament.

Primerament es necessari fer la descarrega de l'arxiu comprimit *jpcap-x64-master.zip*, aquest es pot trobar a GitHub, a la següent adreça, <https://github.com/jovigb/jpcap-x64>.

Una vegada descarregat, localitzem l'arxiu *jpcapx64.dll*, que es troba a dintre de la carpeta *lib* dintre de l'arxiu comprimit, la ruta de la carpeta és:

jpcap-x64-master.zip\jpcap-x64-master\lib

Aquest arxiu se li ha de canviar el nom per *Jpcap.dll* i copiar-lo en dues ubicacions concretes de la carpeta de sistema Windows que es troben normalment a la unitat C:\, la ubicació d'aquestes és:

C:\Windows\System32

C:\Windows\SysWOW64

Quan es realitzi aquesta operació el sistema ens indicarà que ja existeix un arxiu amb aquest nom, cal substituir el ja existent pel que nosaltres hem canviat el nom. Una vegada copiat l'arxiu *Jpcap.dll* de 64 bits a les ubicacions indicades, ja estarà l'equip preparat per al correcte funcionament de l'aplicació.

Finalment, per tal que la informació *WHOIS* es mostri correctament, cal assegurar-se que l'arxiu *ipv4-address-space.csv*, es troba a la mateixa carpeta que l'executable de *Java* de l'aplicació *GeolocalitzacioIP.jar*, ja que aquest és necessari per fer la consulta del servidor *Whois* corresponent.

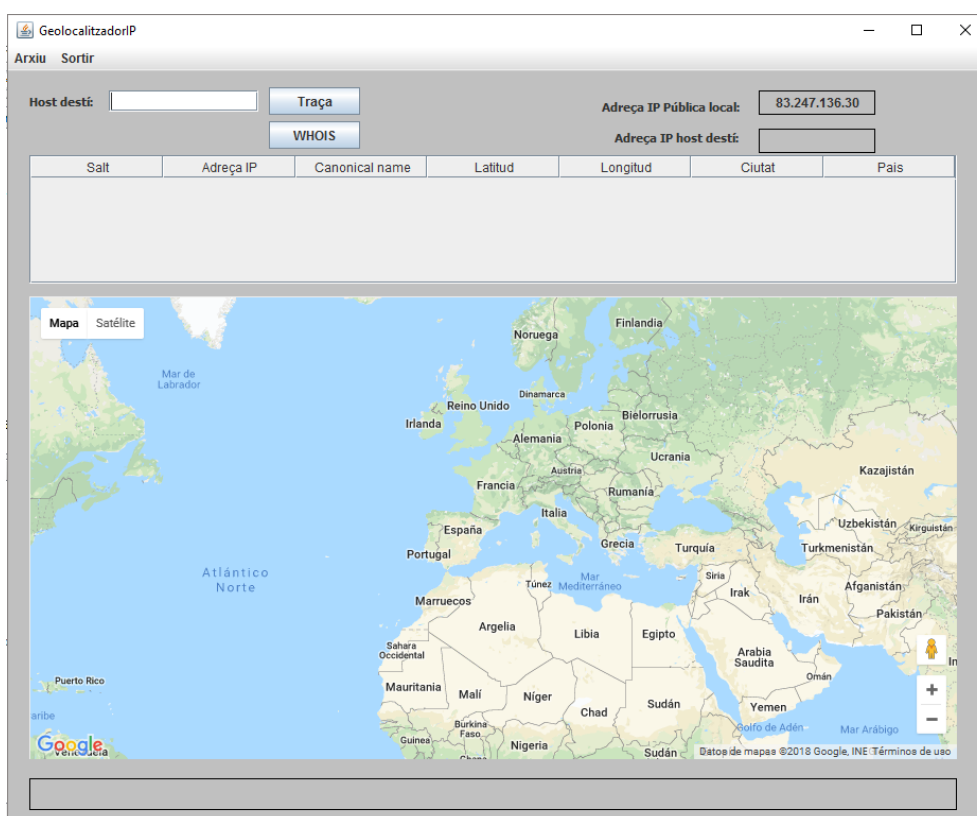
9.2. Annex B. Manual d'usuari.

Una de les característiques d'aquesta aplicació és que el seu funcionament és molt senzill. En aquest annex es farà una breu explicació del funcionament d'aquesta.

Per executar-la, es pot fer des del mateix Windows fent doble clic sobre l'arxiu *GeolocalitzacioIP.jar* que es troba a la carpeta *dist* del projecte *GeolocalitzacioIP*, o des d'una finestra *Ms-dos* situant-nos en la carpeta *dist* i escrivint la següent instrucció:

```
java -jar Geolocalitzacio.jar
```

Una vegada executat el arxiu apareixerà la finestra principal de l'aplicació, on es poden fer totes les accions que aquesta ofereix i on es mostrarà la informació relativa als nodes i el mapa, tal com es mostra a continuació.



En aquesta pantalla inicial es poden veure els elements principals, els botons i els menús d'acció de l'aplicació, a continuació s'enumeren amb una breu explicació de que fa cadascun:

- Menú "Arxiu", conté dos accions que són la d'obrir i guardar una traça per poder accedir a traces anteriors en qualsevol moment. El funcionament d'aquestes opcions és molt senzill, apareix una finestra per navegar per les carpetes de l'equip i poder seleccionar l'arxiu que es vol obrir o indicar el nom del arxiu que es crearà per guardar la traça, segons l'opció triada.
- Menú "Sortir", la seva funció és facilitar el tancament de l'aplicació per a l'usuari.

- Quadre de text “Host destí”, és l’espai on s’ha de escriure el domini o l’adreça IP al qual es vol fer la traça i localitzar al mapa. L’adreça IP ha de ser en format vàlid, és a dir, quatre números decimals separats per un punt d’entre 0 i 255 (XXX.XXX.XXX.XXX). Si no és coneix l’adreça IP del domini destí, es pot escriure el nom del domini, sense les lletres *www* inicials, per exemple *yahoo.es*.
- Botó “Traça”, aquest té la funció d’iniciar el procés una vegada s’ha introduït el domini. Si el domini introduït no és vàlid, apareixerà un missatge d’error indicant-ho. En el moment que comenci la traça de la ruta, aquest botó no estarà operatiu, i apareixerà al costat un altre botó anomenat “Stop”, per tal de poder detenir el procés en cas de necessitat.
- Botó “WHOIS”, proporciona la informació sobre el domini que s’està fent la traça que hi ha als servidors WHOIS. Aquesta informació es mostra en una altra finestra que apareix quan premem el botó i que es pot tancar quan es desitgi. Si no s’ha iniciat cap traça i estem a la pantalla inicial, l’aplicació no retorna cap valor.
- Quadre de text “Adreça IP Pública local”, mostra des de l’inici l’adreça IP local pública de l’equip que executa l’aplicació. Si es carrega una traça anterior feta des d’una altra ubicació, aquest quadre mostra l’adreça IP pública des d’on es va fer aquesta traça.
- Quadre de text “Adreça IP host destí”, es mostra l’adreça IP del equip destí, aquesta es mostra tant si s’ha ficat com a host destí un domini, com si s’ha ficat una adreça IP.
- Taula de nodes, la funció d’aquesta taula, es mostrar els diferents salts que es troben durant la traça i mostra els equips que responen a la consulta ICMP. En el moment que aquesta taula s’emplena apareix un *scroll* al costat dret que permet desplaçar la taula per consultar tots els nodes trobats.
- Mapa, en aquest es mostren els nodes trobats i la traça entre ells. Per tal de fer més visible el resultat, el node inicial s’ha marcat de color verd i el node final de color vermell, els nodes intermedis apareixen de color blau. Al igual que qualsevol mapa de *Google*, aquest permet utilitzar els controls de zoom que apareixen a la part dreta inferior per apropar o allunyar el mapa. Igualment, es pot seleccionar el tipus de vista entre satèl·lit o mapa amb els botons situats a la part superior esquerra del mapa.