IES Palas Atenea

Proyecto de Investigación Bachillerato de excelencia

Programación, Redes y Código Libre

David Davó

Tutor
Julio Sánchez

13 de septiembre de 2016

Índice general

0. Introducción?? 1

1. Programación y código libre 2
1.1. Herramientas . . . 2
1.1.1. GNU/Linux . . . 2
1.1.2. Git y Github . . . 2
1.1.3. LaTeX . . . 4
1.1.4. Python . . . 4
1.1.5. Gtk+ . . . 4
1.1.6. Atom . . . 5
1.1.7. Wireshark . . . 5

2. Redes Informáticas 6
2.1. Capas de Red/Modelo OSI . . . 6
2.2. Topologías de red . . . 6
2.2.1. Clasificación de las topologías de red . . . 7
2.2.2. Nodos de una red . . . 7
2.2.3. Enlaces de red . . . 8
2.3. Paquetes de red . . . 9
2.3.1. Ejemplo: Paquete de red . . . 10
2.4. Protocolos . . . 10
2.4.1. Familia de protocolos de internet . . . 10
2.5. Seguridad de redes . . . 12
2.5.1. Tipos de ataques . . . 12
2.5.2. Contramedidas . . . 13

3. El simulador de redes 15
3.1. Instalación . . . 15
3.1.1. Ubuntu / Debian . . . 15
3.1.2. Arch Linux . . . 15
3.1.3. Ejecución manual / instalación portable . . . 15
3.2. Uso del programa . . . 16
3.2.1. Configuración . . . 17
3.2.2. Dispositivos . . . 18

Glosario y acrónimos 19

A. Unidades de transferencia de datos 20

B. Capturas de pantalla del programa 21

Bibliografía 23

Índice de figuras

1.1. *Gitflow* o flujo de trabajo de Git 3

2.1. Encapsulación de red. El Datagrama IP es lo considerado 'Paquete de red' 10
2.2. Captura de pantalla de Wireshark 10
2.3. Wireshark: SMTP sin encriptación 14
2.4. Wireshark: HTTP Form sin encriptación 14

3.1. Interfaz de InvProy Alpha 16
3.2. Menú de **Información de Dispositivos** junto a una red de topología de malla . . 17

B.1. Captura: Click derecho en un computador 21
B.2. Captura: Ventana para enviar ping. 21
B.3. Captura: Igual que B.2, pero con una IP válida. 21
B.4. Captura: Ventana con la tabla que poseé el Switch. 21
B.5. Captura: Paquetes viajando por una red de ejemplo. 22

Capítulo 0

Introducción??

Internet, El Internet o La Internet consiste en una caja negra misteriosa con un indicador LED rojo que parpadea y contiene todo el poder del mundo en su interior. El internet no tiene cables, es muy ligero e increíblemente pequeño. Internet se encuentra en lo alto del Big Ben (donde hay mejor recepción) y antes de sacarlo de allí debe ser desmagnetizado por un Gran Maestro del Internet (*Elders of the Internet*)como Stephen Hawking, Linus Torvalds o Richard Stallman. [5]

Capítulo 1

Programación y código libre

Propuesta

El objetivo es el desarrollo de un software programado en Python de código libre con el que los alumnos puedan aprender tanto sobre redes como de programación en Python.

1.1. Herramientas

El programa ha sido creado con herramientas de software libre. Según la Free Software Foundation "«Software libre» es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el «software libre» es una cuestión de libertad, no de precio. Para entender el concepto, piense en «libre» como en «libre expresión», no como en «barra libre». En inglés a veces decimos «libre software», en lugar de «free software», para mostrar que no queremos decir que es gratuito." –[4]

Todas las herramientas citadas a continuación, son o están basadas en Software Libre.

1.1.1. GNU/Linux

También llamado incorrectamente sólo Linux, es una manera de llamar al Sistema Operativo (OS) combinación del kernel Linux (Basado en Unix) y el OS *GNU's Not Unix* (GNU no es Unix) (GNU), ambos softwares son libres y de código abierto. Está basado en, y es uno de los ejemplos de, el código libre. Es el sistema operativo más utilizado, ya que la mayoría de los servidores lo usan, y además, otros sistemas operativos como Android están basado en éste.

Distros

Son las distintas distribuciones de software de GNU/Linux. Es decir, un conjunto de software preconfigurado y compilado formado por el Sistema Operativo GNU, el kernel de Linux y otros tantos paquetes, dependiendo de los usuarios a los que esté dirigida esta. Pueden crearse con el soporte de una empresa; como Ubuntu (~~Amazon~~ Canonical Ltd.), openSUSE (Novell) o Fedora (Red Hat); y otras mantenidas por comunidades como Debian, Gentoo o Arch Linux.

He usado dos distros diferentes. Una llamada `Arch Linux`, que es *rolling release* (No tiene "versiones", sino que siempre se va actualizando con los últimos paquetes disponibles, por lo que siempre está actualizado) y otra es Ubuntu 16, basado en Debian, por lo que está bastante menos actualizado y se han tenido que hacer correcciones en el programa para que pueda funcionar con versiones más antiguas de las dependencias.

1.1.2. Git y Github

Git es un software diseñado por Linus Torvalds con el que puedes crear un Sistema de Control de Versiones o VCS (*Version Control System*). Este programa te permite de forma sencilla volver a una versión o *commit* anterior del programa, así como enviarlas a un repositorio remoto

e incluso publicarlas en línea. Su punto fuerte son las *branches* o "ramificaciones"del código, haciendo que la rama *master* (principal) siempre pueda ser usada. Para ello creamos una nueva rama para cada nueva funcionalidad del programa. La implementación del nuevo código a otra rama se denomina *merge*. Otra de las funcionalidades que implementa es clone, que te permite descargar un proyecto si tienes la URL del repositorio git.

Para usar Git, se suele recomendar seguir un *Git workflow* o flujo de trabajo de Git. El más común es el basado en 4 nuevas ramas, a parte de `master`.

Develop: es la rama de desarrollo. Se van aplicando las nuevas funcionalidades a esta rama, para luego convergerlas en la rama `Release` que se va a publicar.

Release: una vez hayamos terminado en la rama de desarrollo, se converge `Develop` con `Release` y se procede a solucionar los bugs que se vayan descubriendo. Cuando se hayan solucionado todos los bugs y la siguiente versión del programa esté disponible para el público, se hace `merge` en `Develop` y en `Master`, además de aplicarle al `commit` una etiqueta con el nombre de la versión. (2.2.1, por ejemplo).

***Hotfix*:** Es una rama dedicada a solventar los bugs que un usuario descubra en una versión ya lanzada de la aplicación. Cuando un usuario descubre un bug, se crea una nueva rama a partir de la última versión de master, se soluciona el bug en esa rama y luego se vuelve a hacer `merge` en master.

***Feature <x>*:** Donde <x>el nombre de la funcionalidad. Es una rama dedicada a una nueva funcionalidad, se crea a partir de `Develop`, y una vez terminada, se hace `merge` en `Develop` de nuevo. Una cosa nueva.

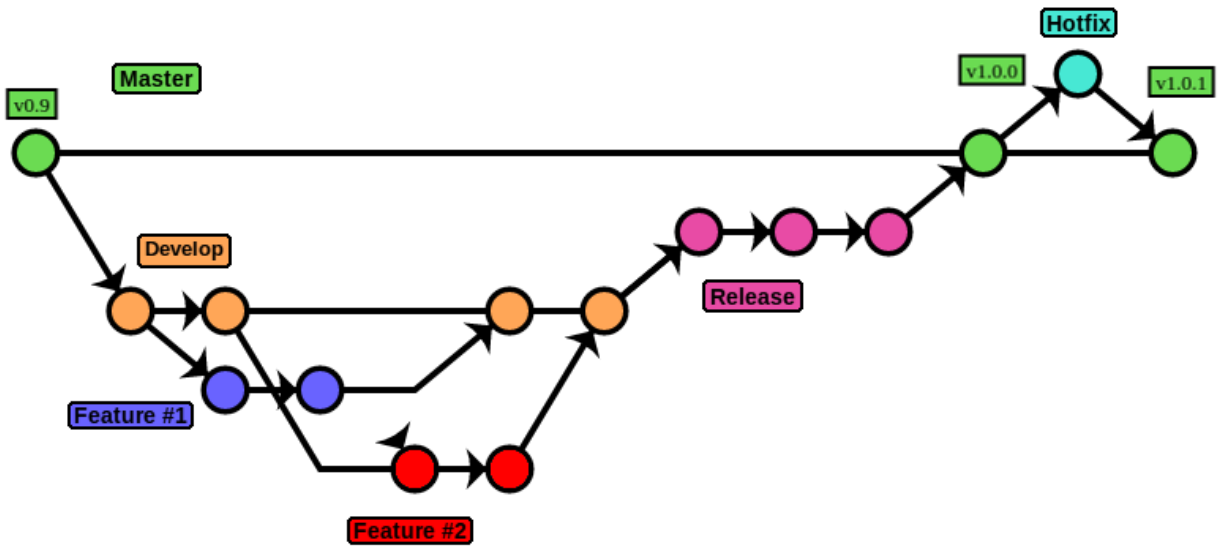


Figura 1.1: *Gitflow* o flujo de trabajo de Git

GitHub es una plataforma de desarrollo colaborativo que te permite alojar tus repositorios Git. Su uso es gratuito si el código almacenado es público. Además, te permite tener una wiki y una página web para tu proyecto, junto a otras funciones. Una de sus funciones estrella es la visualización online del repositorio, con la que cualquier persona tiene acceso al código y los archivos antes de descargarlos. Otra función útil es el apartado de *Issues*, en el que los usuarios de tu código pueden reportar los bugs del programa o aportar nuevas ideas en forma de ”foro”. Tanto el programa como este documento están disponibles en GitHub en los siguientes enlaces. `https://github.com/daviddavo/InvProy` y `https://github.com/daviddavo/InvProy-tex`

1.1.3. LaTeX

LaTeX o, en texto plano, LaTeX, pronunciado con la letra griega Ji (χ), es un software libre orientado a la creación de textos escritos comparable a la calidad tipográfica de las editoriales. Mediante la importación de paquetes y comandos o macros se puede dar formato al texto al igual que con cualquier otro editor, exportándolo posteriormente a PostScript o PDF. Está orientado a documentos técnicos y científicos por su facilidad a la hora de incluir fórmulas o código e importar paquetes que cumplan tus necesidades. No es un procesador de textos, pues está más enfocado en el contenido del documento que en la apariencia de éste. El código del documento puede ser editado con cualquier editor de texto plano como *nano* o *emacs*, aunque he usado una IDE llamada **texmaker**.

1.1.4. Python

Es un lenguaje de programación interpretado (sólo traducen el programa a código máquina cuando se debe ejecutar esa parte del código, por lo que no hace falta compilarlo) que destaca por pretender una sintaxis más legible que la de el resto de lenguajes. Soporta tanto programación imperativa como programación orientada a objetos. Usa variables dinámicas, es multiplataforma, y, además, es de código abierto, lo que permite distribuir el programa en Windows al distribuir los binarios de Python junto a él. En este caso, la versión de Python usada es la 3.4 en adelante.

1.1.5. Gtk+

Es un conjunto de bibliotecas o librerías (conjunto de funciones y clases ya definidas preparadas para el uso de los programadores) desarrollado por la GNOME foundation destinado a la creación de GUIs (Interfaz Gráfica de Usuario), también, al igual que Linux forma parte del proyecto GNU.

Contiene las bibliotecas de GTK, GDK, ATK, Glib, Pango y Cairo; de las que he usado fundamentalmente GTK para crear la interfaz principal del programa; GDK al usarlo como intermediario entre los gráficos de bajo nivel y alto nivel y Cairo para la creación de algunos de los elementos gráficos del programa.

Al usar este conjunto de librerías, he conseguido que sólo sea necesario descargar una dependencia del programa, que además suele venir instalada en la mayoria de distros de Linux. Por ejemplo en una instalación limpia de Ubuntu 16 (sin descargar paquetes adiccionales) el programa funciona perfectamente. Para usarlo en Python se ha tenido que importar la libreria de PyGtk, que también suele venir incluida en la distribución.

1.1.6. Atom

Atom es un editor de código multiplataforma con soporte para plugins escrito en Node.js, también tiene soporte para Git. También es un programa de código libre haciendo uso de la licencia MIT.

1.1.7. Wireshark

Wireshark es un *packet sniffer* o analizador de paquetes. Te muestra los paquetes de red reales enviados y recibidos por una tarjeta de red, lo que facilita la creación del simulador de redes. También te separa las distintas partes de la encapsulación del paquete, además te permite añadir distintos filtros para las distintas capas.

Capítulo 2

Redes Informáticas

Historia

Internet, tal y como lo conocemos ahora, haciendo uso de IPv6, HTML5, CSS3 no existía hasta hace poco, pero el desarrollo de éste transcurre desde los años 60. En 1961 se publican los primeros artículos de Conmutación de paquetes

2.1. Capas de Red/Modelo OSI

El modelo OSI (*Open Systems Interconnection* (Interconexión de Sistemas Abiertos)) es un modelo de referencia para redes basado en capas de abstracción. El objetivo del modelo OSI es conseguir la interoperabilidad entre sistemas con la protocolos estandarizados. Fue creado en 1980 por la ISO (*International Organization for Standardization*). No es considerado una arquitectura de red porque los protocolos no forman parte del modelo, sino son entidades de distintas normativas internacionales.

Capa	PDU[1]	Función	Ejemplos
1. Física	Bit	Transmisión y recepción de bits físicos sobre un medio físico (topología de red)	RJ45, IEEE 802.11, etc.
2. Data Link	Frame	Transmisión segura de *frames* entre dos nodos conectados por una capa física.	Ethernet, 802.11, etc...
3. Red	Paquete	Estructurar y administrar una red multinodo. Incluye enrutamiento, control de tráfico, y asignación de direcciones	IPv4, IPv6, ICMP...
4. Transporte	Datagrama(UDP) Segmento(TCP)	Transmisión de segmentos de datos entre los puntos de una red, incluyendo ACK	TCP, UDP...
5. Sesión	Datos	Administración de sesiones de comunicación, como intercambio continúo de información entre dos nodos.	SSH, RPC, PAP...
6. Presentación	Datos	Translación de datos entre un servicio de red y una aplicación. Incluye comprensión, encriptación/decriptación, y codificación de carácteres.	MIME, TLS
7. Aplicación	Datos	APIs de alto nivel, incluyendo recursos compartidos y acceso remoto de archivos	HTTP, FTP, SMTP...

2.2. Topologías de red

La topología de red es la configuración de los elementos que componen una red. Puede ser representada lógica o físicamente. La topología lógica puede ser igual en dos redes, aunque su topología física (distancia entre conexiones, tipo de señales...) pueda ser distinta. Se distinguen

[1] *Protocol Data Unit* o Unidad de Datos de Protocolo.

dos elementos: los nodos (Ordenadores, switches, etc.) y los enlaces (medio de transmisión de los datos).

2.2.1. Clasificación de las topologías de red

Se distinguen ocho tipos de topologías de red: [1]

Punto a punto: conexión directa entre los dos puntos de la red. También es conocida como *P2P* (*Peer to Peer*).

Estrella: cada host se conecta a un hub central con una conexión P2P. Cada nodo está conectado a un nodo central que puede ser un router, hub o switch.

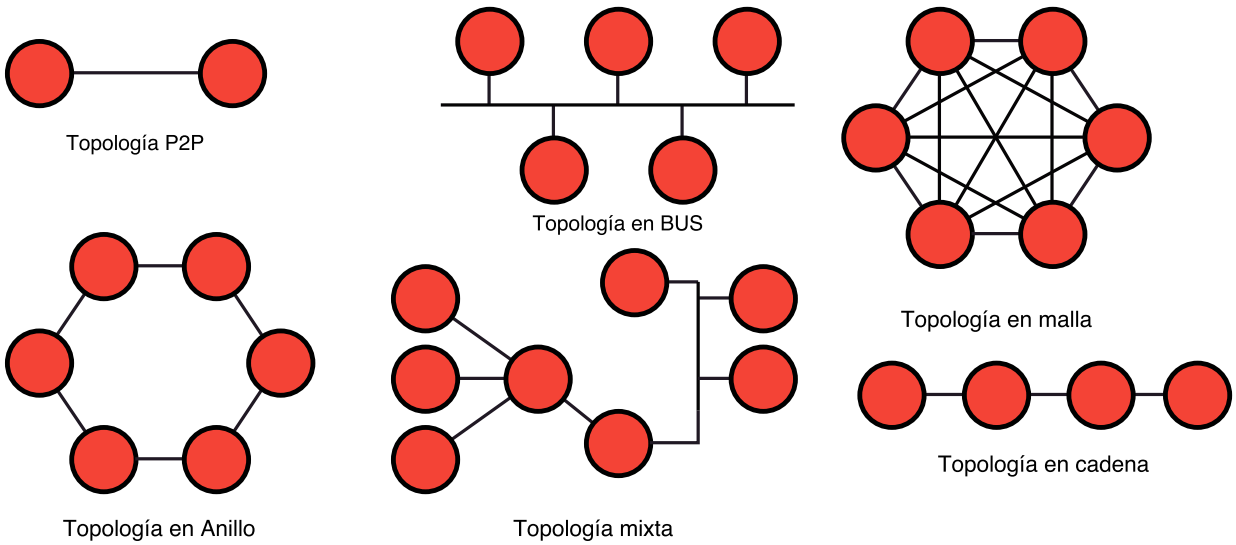
Bus: cada nodo está conectado a un sólo cable. Una señal de un dispositivo viaja en ambos sentidos por el cable hasta que encuentra el destino deseado.

Anillo: es una topología en bus pero con los extremos conectados. Los datos atraviesan el anillo en una única dirección y van atravesando cada uno de los nodos, por lo que si uno de ellos no funciona, la red tampoco.

Malla: se pueden distinguir dos tipos: completamente conectados, en la que todos los nodos están conectados entre ellos y parcialmente conectados, en la que algunos nodos pueden estar conectados punto a punto y otros pueden tener varias conexiones.

Híbrida: combinan dos o más topologías. La más famosa es la topología de **árbol**, en la que se conectan varias topologías de estrella mediante bus.

Cadena: se conecta cada ordenador en serie con el siguiente. Cada ordenador repite el mensaje al siguiente ordenador si éste no es su destino. Si se cierra el circuito se crea una topología en anillo, mientras que si se deja abierto se denomina topología linear.



2.2.2. Nodos de una red

***Router* o enrutador:** es un dispositivo de red que reenvía los paquetes mirando en la capa 3 del modelo OSI (IP) y conecta dos redes.

Puente de red o *bridge*: Funciona en la capa 2 del modelo OSI. Es un dispositivo que conecta dos segmentos de red formando una única subred, por lo que las dos "redes"pueden conectarse e intercambiar datos sin necesidad de un *router*.

Conmutadores o *switches*: dispositivo de red que filtra los datagramas del nivel 2 OSI (*Data Link Layer*, ver 2.1, pág. 6), también conocidos como *frames*, y reenvía los paquetes recibidos entre los puertos, dependiendo de la dirección MAC de cada *frame*. La diferencia entre un *switch* y un *hub* es que el *switch* sólo reenvía los paquetes por el puerto necesario. También existen un tipo especial de *switches* que pueden mirar en el nivel 3 OSI.

Repetidores y hubs: un repetidor es un dispositivo de red que, llegada una señal, limpia el ruido innecesario y la regenera. Un repetidor con múltiples puertos es un hub, trabajan en la capa 1 del modelo OSI (*Open Systems Interconnection* (Interconexión de Sistemas Abiertos)). Los repetidores requieren un pequeño tiempo para regenerar la señal, lo que puede crear un retardo en la señal.

Interfaces de Red: también conocido como tarjeta de red o *Network Interface Controller* (NIC), es un hardware, normalmente integrado en la placa base, que permite al ordenador conectarse a una red. Recibe el tráfico de una dirección de red. En las redes de Ethernet, tiene una dirección MAC (*Media Access Control* [Control de Acceso al Medio]) única. Estas direcciones son administradas por el IEEE (Instituto de Ingeniería Eléctrica y Electrónica) evitando la duplicidad de estas. Cada dirección MAC ocupa 6 octetos, o 48 bits, a lo que suele ser representada como una cadena hexadecimal, por ejemplo: "43:31:50:30:74:33".

Módem: Dispositivos que transforman señales analógicas a digitales y viceversa. Son usados mayoritariamente en el ADSL (*Asymmetric Digital Subscriber Line* [Línea de Abonado Digital Asimétrica]).

Cortafuegos o *firewalls*: dispositivo que controla la seguridad mediante reglas de acceso. Aceptan determinados paquetes mientras rechazan otros. En una red doméstica, se puede poner un firewall que sólo acepte tráfico de los puertos de uso común (Páginas Web, e-mail, etc.) y rechace otros más peligrosos (Acceso remoto, SSH, SMTP, SOCKS...).

2.2.3. Enlaces de red

Según el modelo OSI, los enlaces de red corresponden a las capas 1 y 2. El medio físico puede ser tanto ondas de radio (Wi-Fi), como fibra óptica (FTTH) o impulsos de red (PLC, Ethernet, DSL).

Cableado

Coaxial: Cables de cobre o aluminio recubiertos de aislante, rodeado de un conductor, así se reducen las interferencias y la distorsión. Normalmente son usados para la transmisión de radio y TV, pero pueden ser usados para redes informáticas. Pueden llegar hasta a 500 Mbit/s **<INSERTAR IMAGENES>**

Par trenzado o *Ethernet*: Es el más usado en redes locales. Es un cable formado por finos cables trenzados en pares. En telefonía se usa el RJ11 o 6P4C (6 posiciones, 4 conectores) formado por 2 pares. Para ordenadores, según el estándar *Ethernet* se usa 8P8C o RJ45 de 4 pares, debido al nombre del estándar, este cable suele ser comúnmente llamado çable de Ethernet". Puede llegar hasta 10 Gbit/s

Fibra óptica: Hilo de cristal o plástico flexible que permite que la luz se refleje en su interior, transmitiéndola de un extremo a otro del cable. No tienen apenas pérdida por distancia y son inmunes a las interferencias electromagnéticas. Además, permiten varias frecuencias de onda, lo que equivale a una transferencia de datos más rápida. Son usados para salvar las largas distancias entre continentes.

Comunicación inalámbrica o *Wireless*

Microondas terrestres: Transmisores, receptores y repetidores terrestres que operan en frecuencias de entre 300 MHz y 300 GHz de propagación de alcance visual, por lo que los repetidores no se separan más de 48 km.

Comunicación satelital: Microondas y ondas de radio que no sean reflejadas por la atmósfera terrestre. Los satélites mantienen una órbita geosíncrona, es decir, el periodo de rotación es el mismo que el de la tierra, lo que se produce a una altura de 35786 km.

Celular o PCS: Ondas electromagnéticas de entre 1800 y 1900 MHz. Son las usadas por los teléfonos móviles. A partir del 2G o GPRS, se podia acceder a Internet con de TCP/IP. El sistema divide la cobertura en áreas geográficas, cada una con un repetidor. Repiten los datos entre un repetidor y el otro.

Ondas de radio: Ondas de 0.9, 2.4, 3.6, o 5 GHz. El estándar más usado es el *IEEE 802.11*, también conocido como wifi o Wi-Fi que opera en la banda de 2.4 GHz, a excepción de la versión IEEE 802.11ac que opera a 5GHz que tiene menos interferencias, pero también menor alcance.

2.3. Paquetes de red

Es cada serie de bits en la que se divide la información enviada por una red. Según el modelo OSI, un paquete es estrictamente el PDU de la capa de red. El paquete de red se encuentra encapsulado en la capa anterior del modelo OSI. Por ejemplo, en éstandares de comunicación TCP/IP, un segmento TCP puede ser llevado por varios paquetes IP transportados por varios frames de Ethernet .Está formado por varios protocolos y en él se distinguen tres partes:

Header o cabecera: Datos e información sobre el paquete. (Dirección IP, MAC, versión, etc)

Payload o carga: Los datos que se quieren transferir.

Trailer o cola: En ocasiones es inexistente (como en UDP) pero suele ser un código de comprobación de errores.

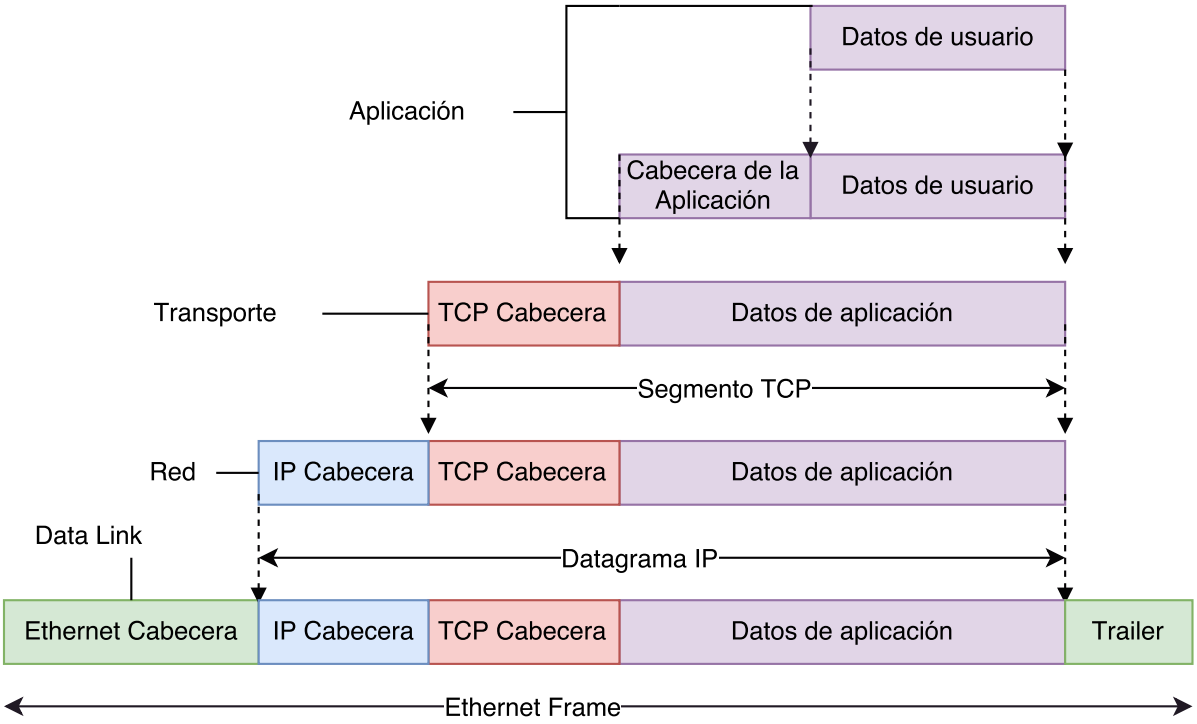


Figura 2.1: Encapsulación de red. El Datagrama IP es lo considerado 'Paquete de red'

2.3.1. Ejemplo: Paquete de red

2.4. Protocolos

Un protocolo de comunicación es un conjunto de reglas para intercambiar información entre enlaces de red. En una pila de protocolos, cada protocolo cubre los servicios del protocolo de la capa anterior. Por ejemplo, un e-mail se envía mediante el protocolo POP3 (*Post Office Protocol*, Protocolo de Oficina Postal) en la capa de Aplicación, sobre TCP en la capa de transporte, sobre IP en la capa de Red, sobre Ethernet para la capa *Data Link*.

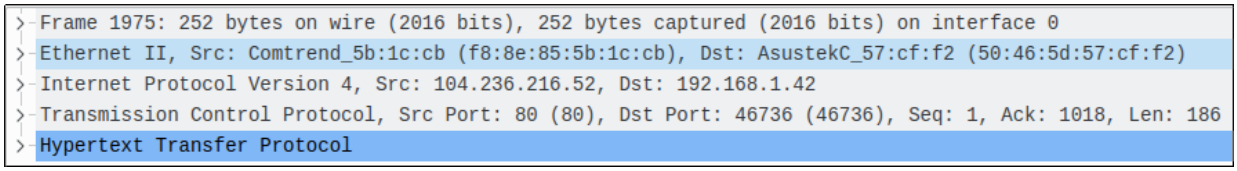


Figura 2.2: Captura de pantalla de Wireshark (Véase 1.1.7, pg. 5) en la que se muestran los protocolos que forman un paquete de red HTTP.

2.4.1. Familia de protocolos de internet

También conocido como *Internet Protocol Suite*, y más conocido como TCP/IP, es el fundamento de las redes informáticas. Se trata de un conjunto de más de 100 protocolos que permiten la conexión de ordenadores tanto en Internet como en LAN, incluyendo protocolos de las aplicaciones más usadas.

Aplicación

Es la capa en la que se envían los datos a otras aplicaciones en otro ordenador o en el mismo. Las aplicaciones hacen uso de las capas inferiores para asegurarse que los datos lleguen a su destino. Algunos de los protocolos más usados son:

- **HTTP** *Hypertex Transfer Protocol:* Protocolo de Transferencia de Hipertexto. Es el protocolo base de la World Wide Web. Se trata de texto estructurado que usa hiperenlaces entre nodos que también contienen texto. El cliente, al entrar en una URL (*Uniform Resource Identifier*, Identificador de Recursos Uniforme), el agente de usuario (navegador) envía al servidor una petición de la página web, mediante HTTP. El servidor, envía como respuesta un documento HTML u otro recurso.
- **DNS** *Domain Name System:* Sistema de Nombres de Dominio. Un servidor DNS almacena una base de datos distribuida y jerárquica con información sobre el nombre del dominio y la dirección IP a la que está vinculada. Al intentar conectar a `http://www.4chan.org`, el cliente pregunta al servidor cual es la dirección IP asociada a esa dirección, y se conecta a tal IP, en este caso 104.16.66.203. Para evitar tener que consultar continuamente con el servidor, se almacenan en una caché en el cliente.
- **TLS/SSL** *Transport Layer Security*, y su predecesor *Secure Sockets Layer*. <VER APARTADO DE SEGURIDAD>
- **HTTPS** HTTP Seguro. Es HTTP con TLS aplicado.
- **DHCP** *Dynamic Host Configuration Protocol*: Protocolo de configuración dinámica del host. Este protocolo es controlado por un servidor DHCP que envía parámetros de configuración automática a los clientes. El ejemplo más común es el de cualquier Router doméstico, que asigna automáticamente a cada dispositivo una dirección IP diferente, pero dejando un rango en el que se pueden establecer IP's estáticas.
- **FTP** *File Transfer Protocol:* Protocolo de Transferencia de Archivos, te permite enviar archivos entre un cliente y un servidor. El protocolo TLS aplicado a FTP se denomina FTPS. Te permite acceder, mediante un usuario y contraseña, o de forma anónima, a un sistema de archivos jerárquico con nombres de archivo codificados. Utiliza el puerto 21 de forma predeterminada.
- **SSH** *Secure Shell:* Terminal seguro. Es un protocolo de red criptográfico que permite a un cliente conectarse a un servidor y ejecutar comandos de terminal como un usuario (conociendo el usuario y contraseña). Además, permite la creación de túneles, lo que permite asegurar cualquier aplicación a través de SSH, y el acceso a puertos bloqueados por el cortafuegos en el cliente. La mayoría de servidores de SSH incluyen un servidor de SFTP, el protocolo FTP con SSH aplicado.
- **IMAP** *Internet Message Access Protocol:* Protocolo de acceso a mensajes de Internet. Usa una conexión TCP/IP para conectarse a un servidor de e-mail y ver el contenido de los mensajes, sin necesidad de descargarlos. A diferencia de POP, te permite usar una bandeja de entrada desde varios clientes.

DHCP, DNS, FTP, HTTP, IMAP, POP, TLS/SSL, SMTP, RIP, SSH, Telnet

Transporte

- **TCP** *Transmission Control Protocol:* Protocolo de Control de Transmisión. Se aplica a los paquetes para administrarles un orden y un sistema de comprobación de errores. Con todas las funcionalidades, ocupa bastante espacio, lo que aumenta la latencia, aunque es más fiable para el envío de la mayoría de los datos.
- **UDP** *User Datagram Protocol:* Es un protocolo muy minimalista. A diferencia del TCP, no garantiza que los paquetes lleguen, o lleguen en orden, o protección ante duplicados. Reduce mucho la latencia ya que no usa *handshaking*. Por ello es usado por ejemplo para *streamings* de televisión o videollamadas.

Red

- **IP** *Internet Protocol*: Protocolo de Internet. Envía datagramas o paquetes de red a través de redes. Tiene una función de enrutamiento que es la que permite la interconexión de redes, y la existencia de Internet. Es un protocolo que encapsula el paquete definiendo en el *header* (cabecera) las direcciones IP del servidor y el cliente, o remitente y destinatario. La versión usada actualmente es IPv4 desarrollado en 1981, pero poco a poco se va abriendo paso la versión IPv6. La mayor diferencia es que la versión cuatro cuenta con direcciones de 32 bits lo que permite tan sólo unas 4.3 millardos (2^{32}) de direcciones, mientras que la versión 6 tiene direcciones de 128 bits, lo que permite más de 340 sextillones (2^{128})de direcciones
- **ICMP** *Internect Control Message Protocol*: Es un protocolo que no es usado por aplicaciones de usuario (a excepción de herramientas de diagnóstico como ping o traceroute). Lo usan los dispositivos de red, como los routers, para enviar notificaciones o mensajes de error indicando que un servicio no está disponible.

Link

- **ARP** *Address Resolution Protocol:* Protocolo de resolución de direcciones. Es un protocolo que convierte direcciones de la capa de Red a la capa de Enlace (dir. IP a dir. MAC).

ARP, MAC, ETHERNET

2.5. Seguridad de redes

La seguridad de redes consiste en el conjunto de acciones que toma el administrador de redes para prevenir y evitar acceso no autorizado, mal uso, o caída del servicio de red.

2.5.1. Tipos de ataques

Hay dos tipos de ataques de red. Son ataques pasivos cuando el intruso intercepta los datos que viajan por la red, y se considera activo cuando el atacante modifica el funcionamiento normal de la red. Aquí algunos ejemplos de los ataques más comunes:

- **Ataques pasivos**
  - **Sniffing o analizador de paquetes:** Mediante un software se muestran los datos de los paquetes de red enviados y recibidos por la red.

- **Escáner de puertos:** Se envían numerosas peticiones al servidor por los servidores más comunes, así se comprueba que puertos están abiertos. Por ello es recomendable cambiar los puertos por defecto de los servidores importantes.
- **Escáner IDLE:** Se realiza un escáner de puertos para saber que servicios están disponibles, pero a traves de otro ordenador "zombie", y observando el comportamiento de éste.

• **Ataques activos**

- **Ataque de Denegación de Servicio:** Se "desborda.[el] ancho de banda mediante el envío de muchas peticiones a un servidor, además de ser de un tamaño excesivo.
- **Ataque DDoS:** *Distributed Denial of Service*, o un ataque de Denegación de Servicio distribuido. Varios ordenadores hacen un ataque DoS a un mismo servidor, algunas veces los ordenadores forman parte de una botnet, y en ocasiones ocurre sin querer (al haber demasiado tráfico de red).
- **Phishing:** Con el objetivo de obtener información como nombres de usuario y contraseña o tarjetas de crédito, se crea una página de apariencia parecida a la página que trata de simular. Los usuarios más incautos no notarán el cambio e introducirán sus datos en esta página.
- **SQL Injection:** Es una técnica de inserción de código. Al pedir un servidor SQL datos como "Nombre" o "Apellido", se introduce junto a estos código malicioso que el servidor puede ejecutar. Por ejemplo, `SELECT * FROM alumnos WHERE nombre = '<nombreintroducido>' ;`. <nombreintroducido>puede ser `Pablo` o `Juan`, pero si se introduce `x' ; DROP TABLE alumnos; SELECT * FROM asignaturas WHERE 't' = 't'`, el código que interpreta el servidor eliminaría la tabla `alumnos` por completo.
- **Ataque Smurf:** Es una especie de ataque DDoS. Se envían paquetes ICMP (probablemente pings) a distintas máquinas, pero estos paquetes que se envían, el valor de la dirección IP del remitente es la dirección IP del objetivo al que se quiere atacar. Por lo que, las máquinas a las que se las ha enviado el mensaje ICMP responderán todas al objetivo, haciendo así un DDoS.
- **DNS poisoning:** Se modifica la caché de DNS de un ordenador, redireccionando a una IP incorrecta, de esta manera se puede realizar un ataque de phishing sin que lo sepa el usuario del ordenador. En el caso de hacerlo con las tablas de ARP, se denomina *ARP Poisoning*.

2.5.2. Contramedidas

Encriptación

Se suele denominar también E2EE o *End-to-end encryption*, es decir, encriptación de punto a punto. Se suelen usar claves PGP (*Pretty Good Privacy*, Privacidad bastante buena) para cifrar correos electrónicos y otros archivos. Para HTTP lo más común es la encriptación TLS, aunque también se está utilizando actualmente para email. El servidor genera o contiene una clave o certificado, luego el cliente, debe recibir o tener esa clave para poder desencriptar el mensaje.

Cortafuegos

Primero necesitamos definir lo que es un **puerto**. Un puerto es un punto final de comunicación en un Sistema Operativo. El puerto siempre está asociado a una dirección IP y a un tipo de

protocolo. Así completa el origen o destino de un paquete de red. Se aplica en la capa de transporte del modelo OSI. El puerto es un número de 16 bits, por lo que será un número comprendido entre 0 y 65536. Multitud de puertos están ya reservados por diversos protocolos y programas, como el 80 para HTTP, 22 para SSH o 25 para SMTP.

Un cortafuegos es un software que supervisa el tráfico de entrada y salida de datos, basado en unas reglas. Si un paquete de red cumple esas reglas, es rechazado. Pueden bloquear un paquete destinado a un puerto, de un protocolo (Bloquear SSH de Internet, pero no local), de una IP específica, entre otros atributos. También pueden configurarse en modo negativo o whitelist, aceptando tan sólo los paquetes que cumplan las reglas. Por ejemplo, puedes especificar que no acepte tráfico en el puerto 23. Pero igualmente puedes especificar que sólo acepte tráfico en el puerto 23.

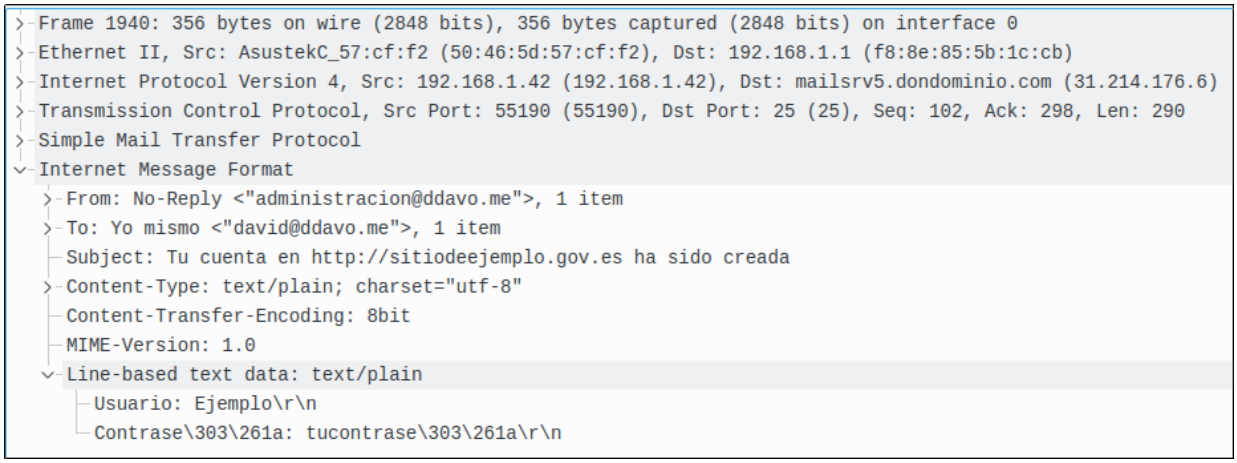


Figura 2.3: Captura de pantalla de Wireshark (Véase 1.1.7, pg. 5) en la que se muestra un paquete SMTP (email enviado) sin ningún tipo de encriptación. Se puede acceder a este paquete desde cualquier nodo de la red.

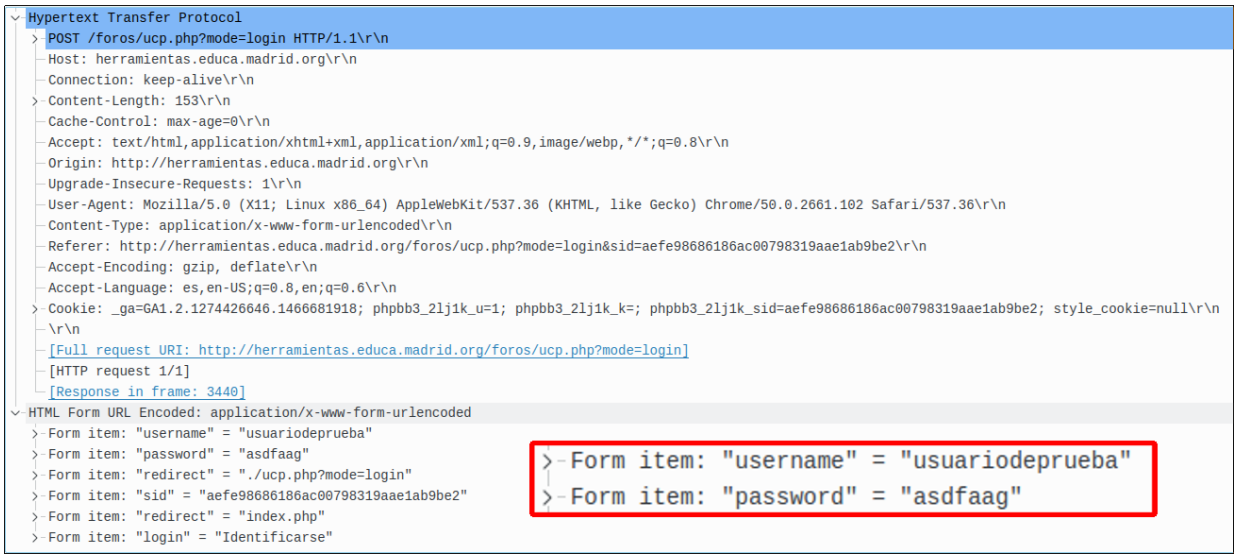


Figura 2.4: Otro ejemplo de captura de paquetes. Esta vez de un formulario de HTTP en el que personas autorizadas podrían ver el usuario y la contraseña.

Capítulo 3

El simulador de redes

3.1. Instalación

3.1.1. Ubuntu / Debian

Tan sólo se debe descargar el paquete del programa. Para ello usa apt-get:

```
~ $ sudo apt-get install invproy
```

En caso de no estar en los repositorios, hay que hacerlo manualmente. Descarga el paquete de `https://github.com/daviddavo/InvProy/releases/latest`. Una vez descargado, abre una terminal donde se haya descargado el paquete e instálelo.

```
Descargas $ sudo dpkg -i invproy_x.y.z_all.deb
```

Donde 'x', 'y', y 'z' son la versión del paquete descargado. Para iniciar el programa debes usar la lista de programas de tu escritorio.

3.1.2. Arch Linux

Puedes encontrar el programa en el AUR <ENLACE>, pero si nunca has instalado nada desde el AUR, debes seguir el siguiente procedimiento.

```
~ $ sudo pacman -S base-devel #Lo necesitas para compilar el paquete
#Ahora elige el sitio donde descargaras el paquete. Aqui no se va a instalar.
~ $ cd Builds
Builds $ curl -O <url> #Lo descargamos
Builds $ tar -xvzf invproy.tar.gz
Builds $ cd invproy
Invproy $ makepkg -sri
```

Y ya lo tendrías instalado en tu ordenador.

3.1.3. Ejecución manual / instalación portable

Lo primero que necesitará es descargar las dependencias. Esto depende del Sistema Operativo. En el caso de GNU/Linux, sólo es necesario descargar `python3-gobject`. Después, clonamos el repositorio de git. Ejemplo en Ubuntu:

```
~ $ sudo apt-get update && sudo apt-get upgrade
~ $ sudo apt-get install git python3-gobject
~ $ cd Descargas
Descargas $ git clone https://github.com/daviddavo/InvProy.git
```

Una vez ya tenemos el repositorio de git clonado:

```
Descargas $ cd InvProy
Descargas $ python3 Main.py
```

En el caso de querer usar el programa con una interfaz gráfica, vamos con nuestro explorador de archivos a la carpeta donde queramos descargarlo. Abrimos una terminal y descargamos el programa con `git clone https://github.com/daviddavo/InvProy.git`. Luego entramos en la carpeta y ejecutamos el archivo `Main.py`

3.2. Uso del programa

Nota: Esta guía ha sido creada usando la versión v0.2.3-alpha, por lo que algunos apartados pueden haber cambiado en versiones posteriores.

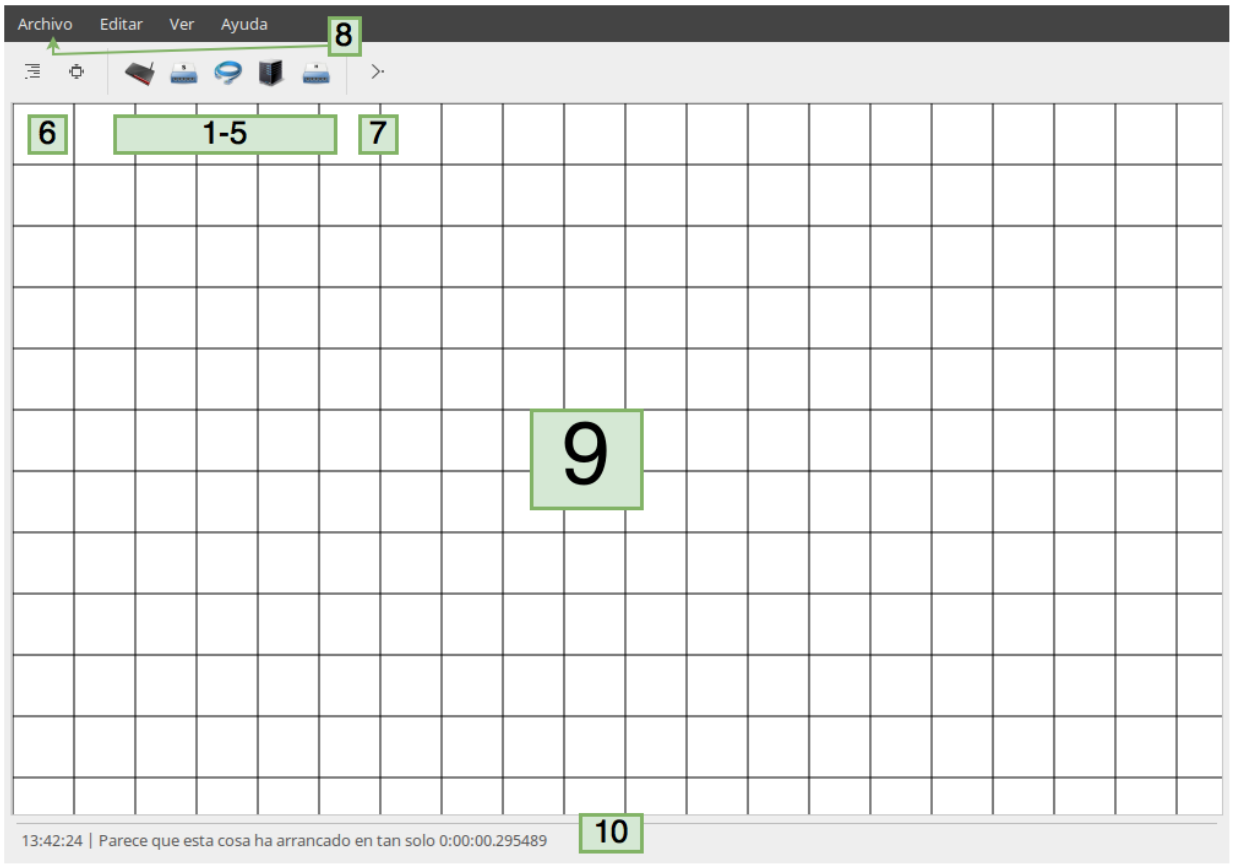


Figura 3.1: Interfaz de InvProy Alpha. Al usar Gtk+, los temas se pueden cambiar, así que la apariencia del programa puede ser distinta dependiendo del tema de escritorio que estés usando. [1]

1-5. También se puede activar con las letras Q, W, E, R, T; respectivamente. Los botones, te permiten (de izquierda a derecha): colocar un router, colocar un switch, conectar dos objetos, colocar un ordenador y colocar un hub.
6. Abre el menú de İnformación de dispositivos", que proporciona información como la dirección IP y MAC, el nombre, o los dispositivos a los que está conectado. (Ver figura 3.2
7. Te permite enviar un ping de un ordenador a otro (El botón funciona a partir de v0.3).
8. Abre el menú de archivo, en el que puedes cargar un archivo, crear uno nuevo, guardarlo, y cerrar el programa.
9. Es la ventana donde puedes colocar los objetos. Puedes moverte a través de ella y en el menú de 'Ver' puedes cambiar el que se vea la rejilla de fondo.

10. Aquí se encuentra una barra con información sobre el funcionamiento actual del programa.

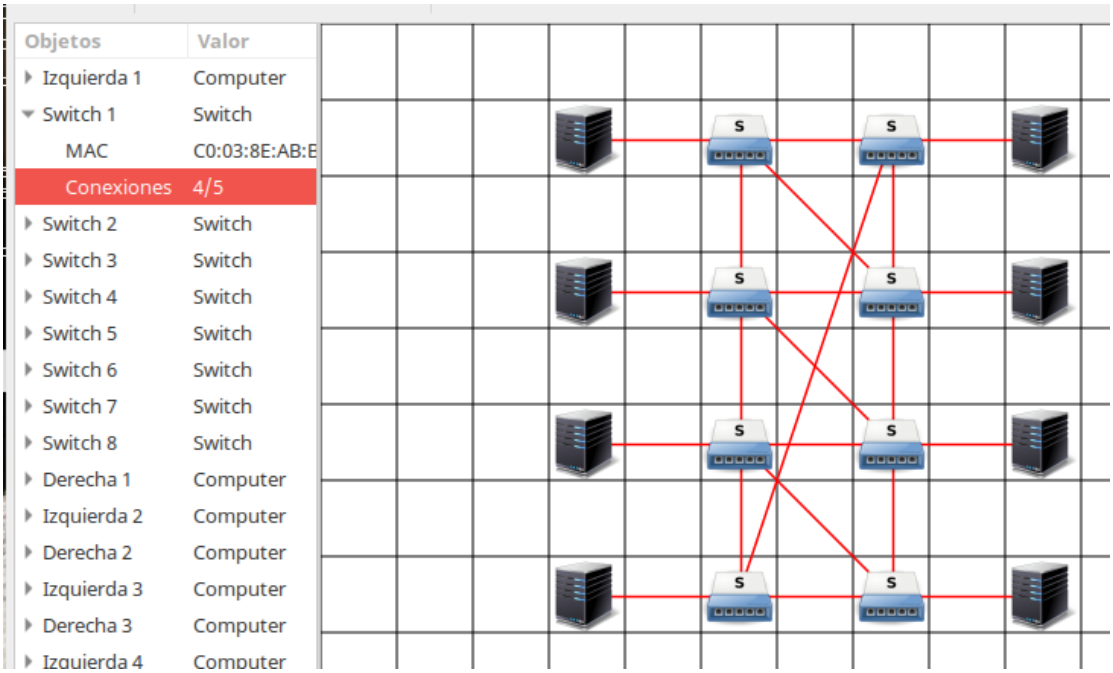


Figura 3.2: Menú de **Información de Dispositivos** junto a una red de topología de malla

Para incluir un objeto en la rejilla, se hace click en el icono del objeto y luego en el lugar donde se quiera poner. Cuando tenemos dos objetos podemos conectarlos si hacemos click primero en el icono del cable, luego en un objeto y después en otro.

Para poder enviar un paquete entre dos Computadores, necesitaremos que estén conectados mediante nodos (Hubs o Switches).

3.2.1. Configuración

Al no haber una ventana de configuración del programa, la configuración debe hacerse de forma manual editando el archivo `Config.ini`. Este es un archivo de texto sin formato en el que se le asigna un valor a cada variable.

`wres` y `hres`: El tamaño (en píxeles) del ancho y el alto de la ventana principal.

`viewport-sqres`: El tamaño en píxeles del lado de los cuadrados de la rejilla.

`viewport-wres` y `viewport-hres`: El número de cuadrados que tendrá de alto y de lancho la rejilla.

`cable-color`: Color por defecto de los cables en HTML.

`start-centered`: Al iniciar el programa, iniciar en el centro de la rejilla en lugar de arriba a la izquierda.

`revealer-show-default`: (True o False). Mostrar por defecto la ventana con la información sobre los dispositivos.

`respack`: Directorio del “Pack de recursos”

`routing-ttl`: Tiempo de vida en segundos de las entradas en la tabla de redireccionamiento de los switches.

3.2.2. Dispositivos

Existen cuatro tipos dispositivos: los Computadores, que tienen la mayor programación; los Switches, que se encargan de manejar los paquetes de red; los Hubs, que son como los Switches, pero reenvían los paquetes por todos sus puertos y los Routers, que tan sólo existen de forma visual, pero no tienen ninguna función de momento. Por lo que sólo vamos a hablar de los Switches y los Ordenadores.

Los ordenadores tienen una función especial que es la de reenviar los paquetes de red. Para ello, en el menú emergente que aparece al hacer click derecho en el objeto, hacemos click en la ventana de ping. Para que el paquete llegue al otro computador, ambos deben tener una dirección IP, y estar conectados a la misma red (Ver Fig. B.2 y Fig. B.3).

Glosario y acrónimos

ADSL *Asymmetric Digital Subscriber Line* [Línea de Abonado Digital Asimétrica]

Bit ***Bi****nary digi****t****, o dígito binario. Cada dígito del sistema de numeración binario*

Botnet Grupo de ordenadores coordinados conectados a un maestro mediante un virus. Gracias a este virus se pueden realizar tareas masivas como el envío de SPAM o ataques DDoS

Bug Error en un programa informático.

Caché Almacenamiento temporal de datos con el objetivo de reducir el retardo, la carga de los servidores y el ancho de banda consumido

Capas de abstracción Método de ocultar detalles de implementación de un set de funcionalidades

Conmutación de paquetes Método para enviar datos por una red de computadoras. Se divide el paquete en dos partes, una con información de control que leen los nodos para enviar el paquete a su destino y los datos a enviar

Datos Secuencia binaria de unos y ceros que contiene información codificada

Dependencia De un programa, otro tipo de software necesario para que éste funcione

FTTH *Fiber To The Home* [Fibra hasta el hogar]

FTTx *Fiber to the X*

GNU *GNU's Not Unix* (GNU no es Unix)

Hardware Conjunto de elementos físicos o materiales que constituyen un sistema informático.

IEEE Instituto de Ingeniería Eléctrica y Electrónica

International Organization for Standardization Organización Internacional de Normalización. Compuesta de varias organizaciones nacionales se encarga de la creación de estándares internacionales desde 1947.

ISO *International Organization for Standardization*

LAN *Local Area Network* [Red de Área Local]

Librería En informática, una librería o biblioteca es un conjunto de recursos y fucniones diseñadas para ser usadas por otros programas. Incluyen plantillas, funciones y clases, subrutinas, código escrito, variables predefinidas...

Linux is a generic term referring to the family of Unix-like computer operating systems that use the Linux kernel

MAC *Media Access Control* [Control de Acceso al Medio]

OSI *Open Systems Interconnection* (Interconexión de Sistemas Abiertos)

POP3 *Post Office Protocol*, Protocolo de Oficina Postal

Programación imperativa Las órdenes del programa cambian el estado de este mismo. Por ejemplo, una variable no tiene por que ser declarada con antelación y su valor es modificable. Es la que usa el código máquina de los ordenadores

Repositorio Servidor donde se alojan ficheros o archivos para su descarga

Test Lorem ipsum dolor sit amet

Topología "Rama de las matemáticas que trata especialmente de la continuidad y de otros conceptos más generales originados de ella, como las propiedades de las figuras con independencia de su tamaño o forma." [3][Topología]

Topología de red Configuración espacial o física de la red. (Ver 2.2 pág.6)

URL *Uniform Resource Identifier*, Identificador de Recursos Uniforme

Apéndice A

Unidades de transferencia de datos

Cantidad de datos transferidos por unidad de tiempo. La unidad de tiempo es el segundo y la cantidad de datos puede ser medida en *bits* (bitrate), carácteres/símbolos (*baudrate*) o bytes (8 bits), en ocasiones también se utilizan *nibbles* (4 bits). Para expresar esta velocidad, se suelen usar múltiplos, que pueden ser en base binaria o decimal.

Se usa la “b”para designar los bits, y “B”para los Bytes. Después, se usan los prefijos del sistema internacional cuando es en base decimal, y los prefijos del SI cambiando la segunda sílaba por “bi”(e.g: kilobit / kibibit, kbit/s / Kibit/s) cuando se trata de múltiplos binarios.

Tabla de múltiplos

Unidad	Símbolo	Equivalencia
Kilobit/s	kbit/s o kb/s	1000 bit/s
Megabit/s	Mbit/s o Mb/s	10^6 bit/s o 10^3 kbit/s
Gigabit/s	Gbit/s o Gb/s	10^9 bit/s o 10^3 Mb/s
Terabit/s	Tbit/s o TB/s	10^{12} bit/s o 10^3 Gb/s
Kibibit/s	Kibit/s	2^{10} bit/s o 1024 bit/s
Mebibit/s	Mibit/s	2^{20} bit/s o 1024 Kibit/s
Gibibit/s	Gibit/s	2^{30} bit/s o 1024 Mibit/s
Tebibit/s	Tibit/s	2^{40} bit/s o 1024 Gibit/s
Byte/s	Byte/s	8 bit/s
Kilobyte/s	kB/s	1000 Byte/s o 8000 bits/s
Megabyte/s	MB/s	10^6 Byte/s o 1000 kB/s
Gigabyte/s	GB/s	10^9 Byte/s o 1000 MB/s
Terabyte/s	TB/s	10^{12} Byte/s o 1000 GB/s
Kibibyte/s	KiB/s	1024 Byte/s
Mebibyte/s	MiB/s	2^{20} Byte/s
Gibibyte/s	GiB/s	2^{30} Byte/s
Tebibyte/s	TiB/s	2^{40} Byte/s

Apéndice B

Capturas de pantalla del programa

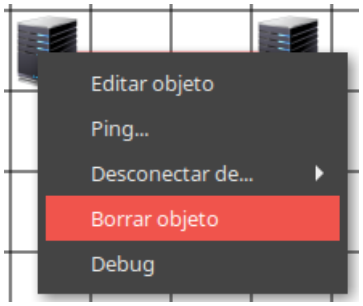


Figura B.1: Captura: Click derecho en un computador

Figura B.2: Captura: Ventana para enviar ping. Está en rojo porque la IP introducida no es válida.

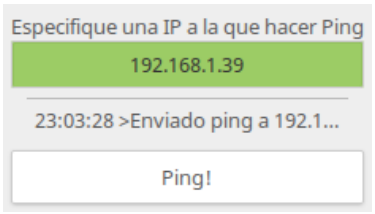


Figura B.3: Captura: Igual que B.2, pero con una IP válida.

Figura B.4: Captura: Ventana con la tabla que poseé el Switch.

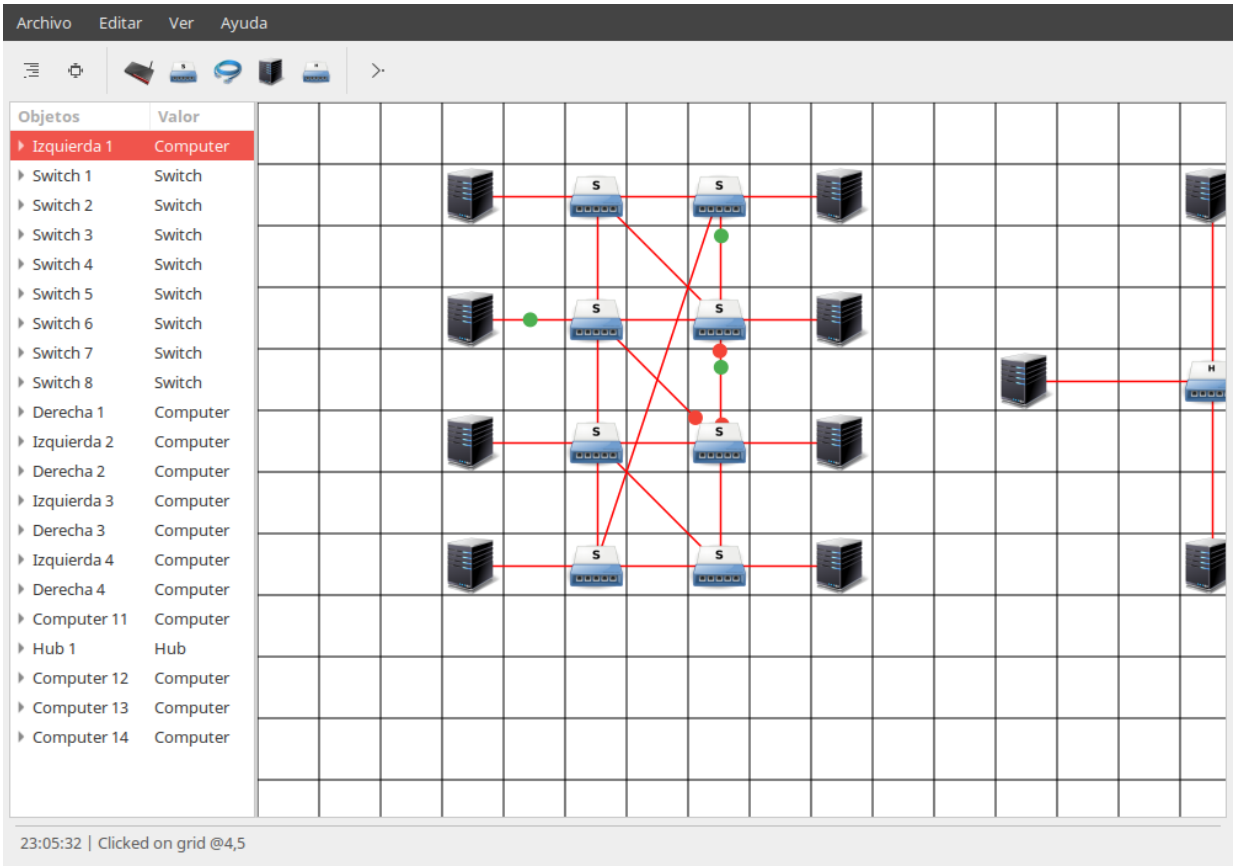


Figura B.5: Captura: Paquetes viajando por una red de ejemplo.

Bibliografía

[1] BICSI. *Network Design Basics for Cabling Professionals*. 2002.

[2] Robert Braden. *RFC 1122*. 1989.

[3] Real Academia Española. *Diccionario de la lengua española, ed. XXIII*. 2014.

[4] FSF. *Filosofía del Proyecto GNU*. 2013. url: https://www.gnu.org/philosophy/philosophy.html.

[5] Roy Trennman & Maurice Moss. *The Internet*. 2009. url: https://www.youtube.com/watch?v=iDbyYGrswtg.

[6] PSF. *What is Python? Executive Summary*. 2016. url: https://www.python.org/doc/essays/blurb/.