

IES Palas Atenea

Proyecto de Investigación Bachillerato de excelencia

Programación, Redes y Código Libre

David Davó

Tutor
Julio Sánchez

23 de agosto de 2016

Índice general

1. Programación y código libre	1
1.1. Herramientas	1
1.1.1. GNU/Linux	1
1.1.2. Git y Github	1
1.1.3. LaTeX	2
1.1.4. Python	2
1.1.5. Gtk+	3
1.1.6. Atom	3
1.1.7. Wireshark	3
2. Redes Informáticas	4
2.1. Capas de Red/Modelo OSI	4
2.2. Elementos físicos de una red	5
2.3. Topologías de red	5
2.3.1. Clasificación de las topologías de red	5
2.3.2. Nodos de una red	5
2.3.3. Enlaces de red	6

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.” –[3]

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

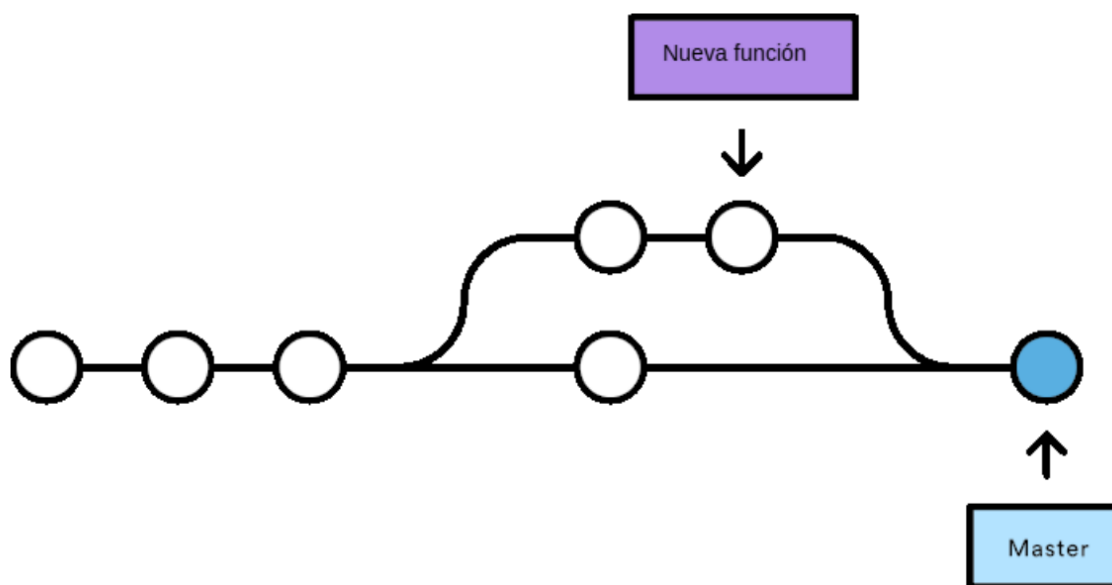
1.1.1. GNU/Linux

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 software son libres y de código abierto. Normalmente Linux se distribuye en distribuciones o 'distros', las cuales contienen paquetes de software preinstalados, dependiendo del grupo de usuarios al que este dirigida.

Distros

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*.

Figura 1.1: *Branching* con 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. Tanto el programa como este documento están disponibles en GitHub en el siguiente enlace. <https://github.com/daviddavo/InvProy>

1.1.3. LaTeX

\LaTeX o, en texto plano, LaTeX, pronunciado con la letra griega Ji (X), 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 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*, pero 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 imperativo como programación orientada a objetos. Usa variables dinámicas, es multiplataforma, y, además, es de código abierto, lo que me 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 mayoría de distros de Linux, por ejemplo en una instalación limpia de Ubuntu 16 (sin descargar paquetes adicionales) el programa funciona perfectamente. Para usarlo en Linux se ha tenido que importar la librería de PyGtk.

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.

Capítulo 2

Redes Informáticas

Historia

Internet, tal y como lo conocemos ahora, haciendo uso de IPv6, HTML5, CSS3 no existe hasta hace recientemente, 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 *Open Systems Interconnection* (Interconexión de Sistemas Abiertos) (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 en sí, sino son entidades de distintas normativas internacionales.

Capa	PDU	Función	Ejemplos
1. Física	Bit	Transmisión y recepción de bits físicos sobre un medio físico	RJ45, IEEE 802.11, etc.
2. Data Link	Frame	Transmisión segura de <i>frames</i> entre dos nodos conectados por una capa física.	Ethernet, 802.11, etc...
3. Red	Paquete	Estructurar y administrar una red multinode. 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 continuo de información entre dos nodos.	SSH, RPC, PAP...
6. Presentación	Datos	Traducción de datos entre un servicio de red y una aplicación. Incluye comprensión, encriptación/decriptación, y codificación de caracteres.	MIME, TLS
7. Aplicación	Datos	APIs de alto nivel, incluyendo recursos compartidos y acceso remoto de archivos	HTTP, FTP, SMTP...

Capa física

En esta capa se encuentra la topología de red (Ver 2.3 pág.5)

2.2. Elementos físicos de una red

Servidor, cliente, switch, hub, router, etc...

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

2.3.1. 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 lineal.

2.3.2. 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. 4), 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...).

2.3.3. 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).

Glosario y acrónimos

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

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

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 funciones 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)

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." [2][Topología]

Topología de red Configuración espacial o física de la red. (Ver 2.3 pág.5)

Bibliografía

- [1] BICSI. *Network Design Basics for Cabling Professionals*. 2002.
- [2] Real Academia Española. *Diccionario de la lengua española*, ed. XXIII. 2014.
- [3] FSF. *Filosofía del Proyecto GNU*. 2013. url: <https://www.gnu.org/philosophy/philosophy.html>.
- [4] PSF. *What is Python? Executive Summary*. 2016. url: <https://www.python.org/doc/essays/blurb/>.

Índice de figuras

1.1. <i>Branching</i> con Git	2
---	---

This work is licensed under a Creative Commons «Attribution-ShareAlike 4.0 International» license.

