

IES Palas Atenea

Proyecto de Investigación Bachillerato de excelencia

# **Programación, Redes y Código Libre**

*David Davó*

Tutor  
Julio Sánchez

20 de agosto de 2016

# Índice general

<b>1. Programación y código libre</b>	<b>1</b>
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+ . . . . .	2
1.1.6. Atom . . . . .	3
1.1.7. Wireshark . . . . .	3
<b>2. Redes Informáticas</b>	<b>4</b>
2.1. Capas de Red/Modelo OSI . . . . .	4

# 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.” –[FSF-Ph]

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

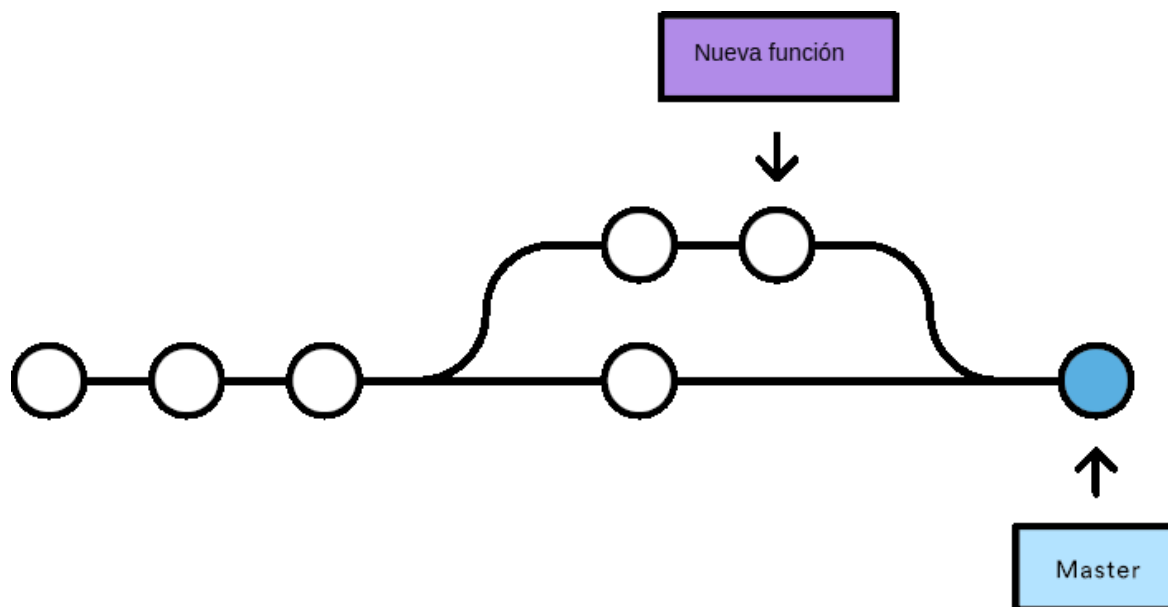
#### 1.1.1. GNU/Linux

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.



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

$\text{\LaTeX}$  o, en texto plano,  $\text{\LaTeX}$ , 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

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+

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 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.

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		Administración de sesiones de comunicación, como intercambio continuo 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 caracteres.	MIME, TLS
7. Aplicación		APIs de alto nivel, incluyendo recursos compartidos y acceso remoto de archivos	HTTP, FTP, SMTP...

# Glosario y acrónimos

**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

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

**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

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

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

