

## Prefacio

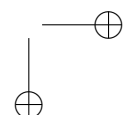
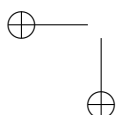
Las redes de comunicaciones, y especialmente Internet, se convirtieron ya hace décadas en una parte esencial de nuestras vidas. Internet cambió radicalmente nuestra forma de comprar, viajar, hacer negocios, y relacionarnos. Fue un cambio mucho más importante que la invención de la imprenta o la revolución industrial, supuso un nuevo concepto de comunicación, un medio de difusión de información y conocimiento a una escala sin precedentes. Y es precisamente la comunicación lo que nos convierte en seres sociales, en humanos. Las redes cambiaron cómo vivimos y lo que somos.

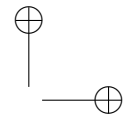
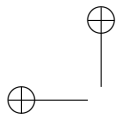
Todo ello es motivo suficiente para que cualquier persona (y en particular cualquier técnico) se interese por el funcionamiento de las redes de computadores. Entender cómo son y cómo funcionan las redes ayuda a tomar conciencia de sus posibilidades y oportunidades, y también de sus riesgos.

### Destripando Internet

Según la RAE una de las acepciones de *destripar* es ‘sacar lo interior de algo’; y eso vamos a hacer aquí, sacar y estudiar las tripas de Internet: la tecnología TCP/IP que la hace posible, los mecanismos y los protocolos clave en su funcionamiento. La mejor manera de conseguir esto es con un enfoque muy práctico, y por suerte, mucha de la tecnología de Internet es sorprendentemente accesible y está disponible para cualquiera. Esto es así en buena medida por la naturaleza abierta de las normas y protocolos que la rigen y por el gran protagonismo del software libre en este ámbito. Basaremos este enfoque práctico en dos tecnologías muy concretas:

- El sistema operativo GNU.
- El lenguaje de programación Python.





## GNU

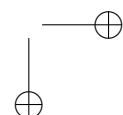
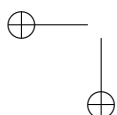
GNU es un sistema operativo estilo POSIX concebido y desarrollado bajo el concepto de software libre. Esto tiene dos implicaciones muy importantes:

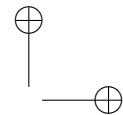
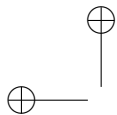
- GNU incorpora todas las ventajas prácticas y técnicas de la tradición de la familia de sistemas Unix. La tecnología base de Internet fue desarrollada sobre Unix, concretamente BSD. Internet y toda la tecnología que la hace posible nació en Unix. GNU es heredero de todo ese legado.
- GNU está disponible para cualquiera, en cualquier parte, sin ninguna limitación. Es la esencia del *software libre*, ideal para toda persona que quiera estudiar el funcionamiento de un sistema completo y productivo con todo detalle. Es tecnología *social* de la que aprender y con la que enseñar, que se puede compartir, mejorar y, por supuesto, crear industria y riqueza. La extensa comunidad de usuarios de GNU y todas sus distribuciones derivadas también es uno de sus puntos fuertes. Los sistemas operativos GNU/Linux y el software libre en general son las opciones más utilizadas con mucho en servidores y en equipos de comunicaciones en todo el mundo.

El entorno de trabajo y herramientas que se utilizan en este libro están disponibles en el sistema operativo Debian GNU/Linux [1]. Por simplicidad y para un máximo aprovechamiento, te aconsejamos instalar esta distribución en tu computador, ya que este será el entorno que asumiremos cuando se hable de configuración o se mencionen comandos del sistema. Todos los programas que emplearemos están disponibles como paquete oficial y se pueden instalar con el gestor de paquetes `apt` sin necesidad de configuración específica.

Por supuesto, cualquier distribución basada en GNU podría ser perfectamente válida para la realización de los ejemplos y ejercicios prácticos propuestos, aunque quizá podrías encontrar ciertas diferencias. Simplemente ten en cuenta, que en ausencia de indicaciones concretas, debes asumir que hablamos de Debian o sus derivados, siendo Ubuntu el más popular.

También asumimos que dispones de un computador con al menos una interfaz de red Ethernet o WiFi con acceso a Internet convenientemente configurada y también que tienes una cuenta de usuario con privilegios de administrador en tu propio computador.





## Python

Python es un lenguaje de programación moderno y muy popular. A pesar de ser un lenguaje muy completo y potente, su aprendizaje resulta sorprendentemente sencillo. Permite, incluso a un programador novato, ser productivo en muy poco tiempo, en comparación con otros lenguajes como Java, C++, o C#. Python ha conseguido una gran notoriedad en los últimos años por su potencia y versatilidad en el manejo de datos (análisis, visualización, etc.) y por su amplio uso en muchas de las tecnologías relacionadas con la inteligencia artificial. Pero Python va mucho más allá.

Una de las características que lo hace especialmente adecuado para la «programación de redes» es que su librería estándar respeta la nomenclatura y conceptos del API de programación de POSIX, lo que resulta muy útil para encontrar documentación y bibliografía, además de facilitar la *traducción* a Python de las llamadas al sistema de POSIX, que están implementadas en C. Cuando en el texto hagamos referencia a una llamada al sistema usaremos el superíndice <sup>sc</sup> ‘*system call*’, como por ejemplo en: `write()`<sup>sc</sup>.

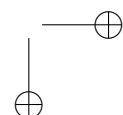
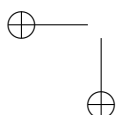
Python también es software libre, lo que significa que está disponible para cualquiera en cualquier plataforma. Está respaldado por una enorme comunidad de usuarios y desarrolladores, y suele ser muy fácil encontrar ayuda, incluso en temas muy específicos.

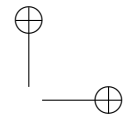
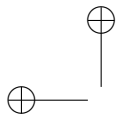
En todos los ejemplos, listados de código y programas que se proporcionan junto con el libro, utilizaremos siempre Python 3.

## Contenido del libro

La mayoría de los libros de redes y comunicaciones suelen tener una estructura similar. Eso se debe a que las redes se estudian, e incluso implementan, mediante los denominados modelos de referencia, especialmente el «modelo OSI». Este modelo divide todo el conocimiento relacionado con las comunicaciones en redes en una serie de capas dispuestas como una pila. La primera (la inferior) es la capa física y se ocupa de los detalles de más bajo nivel llegando a los valores de voltaje o el tipo de cableado. La última es la capa de aplicación y se ocupa del más alto nivel, como por ejemplo, la estructura de un *e-mail*. Los autores suelen seguir este modelo ya sea empezando por la inferior hacia arriba o desde la superior hacia abajo, explicando las tecnologías y funcionamiento de cada componente capa a capa.

El problema con este enfoque es que esas capas no son ni de lejos tan independientes como se dice. Hay muchos mecanismos esenciales que requieren entender los detalles de lo que ocurre en otras capas. A nosotros no nos





parece buena idea ignorarlos o postergar su explicación, que es lo que suele ocurrir. Para evitarlo, en este libro hemos preferido centrarnos en los conceptos y mecanismos de las redes aunque su explicación pueda involucrar varias capas. Pensamos que esto es más adecuado para novatos aunque quizá sea menos formal. En todo caso, no vamos a ignorar los modelos de referencia; son muy útiles para ubicar cada componente y entender cómo se relaciona con el resto. Los mencionaremos con frecuencia, simplemente no los utilizaremos como hilo conductor.

Veamos un pequeño resumen de la estructura general del libro:

### **Shell (C2) y Python (C3)**

introducen dos tecnologías que vas a necesitar para seguir los frecuentes comandos de consola y pequeños programas que se utilizan en todo el libro. La shell y Python son el soporte pragmático del texto, de modo que si no tienes experiencia previa con ellos, es muy conveniente que los leas y practiques los ejemplos que incluyen. Si consideras que tienes suficiente familiaridad con ellos, puedes omitir estos capítulos.

### **Internet (C4)**

aborda algunos de los conceptos más básicos: red, interred o protocolo, y los modelos de referencia, incluido el citado modelo OSI.

### **Protocolos esenciales (C5)**

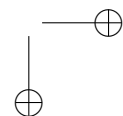
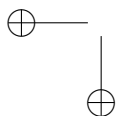
introduce los protocolos elementales de la familia TCP/IP. Aunque solo se cuenta lo mínimo de su funcionamiento, sí se presenta el formato de cada mensaje. Esto es así porque queremos que te pongas manos a la obra lo antes posible, que captures, veas, y entiendas tráfico real sin esperar a leer toda la teoría.

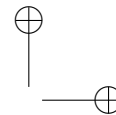
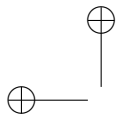
### **Sockets (C6)**

Los *sockets* son la herramienta de programación elemental para escribir software de redes. Aunque muchos autores los consideran un tema avanzado y muchos libros los omiten completamente, aquí incluimos esta introducción muy al principio. Entender lo que el programador puede hacer o no te va a ayudar a comprender todo lo demás.

### **Direccionamiento IP (C7)**

Las redes se utilizan para comunicar computadores —eso seguro que ya lo sabes— y para eso cada computador debe tener una dirección única. Este capítulo explica con detalle cómo se organizan las direcciones IP dentro de una red y cómo se puede saber a qué red está conectado cada computador a partir de su dirección.





### **Interconexión de redes (C8)**

La interconexión de redes es el concepto clave de Internet. Aquí verás cómo los routers son capaces de mover paquetes de una red a otra, conceptos clave como la tabla de rutas, los métodos de entrega de paquetes o cómo se resuelve el problema de la fragmentación de paquetes.

### **Configuración IP (C10)**

Aprenderás cuáles son los datos de configuración esenciales que requiere un nodo para poner conectarse de forma correcta a una red, cómo conseguir esos datos y cómo es posible automatizar el proceso.

### **Encaminamiento dinámico (C9)**

Las redes cambian constantemente, y los routers necesitan adaptarse a esos cambios. Aquí verás los mecanismos que permiten a los routers trabajar juntos para encontrar la mejor ruta en cada momento y así llevar cada paquete hasta su destino.

### **Confiabilidad (C11) y Control de congestión (C13)**

El protocolo de red IP no da ninguna garantía de que los datos vayan a llegar sin errores, ni siquiera que vayan a llegar. En estos dos capítulos verás cómo es posible añadir mecanismos que garanticen la entrega correcta de los mensajes, corrigiendo errores y retransmitiendo mensajes perdidos. También verás qué es la congestión de la red y cómo se combate.

### **Cliente-servidor (C14)**

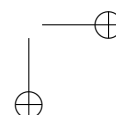
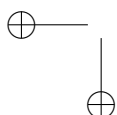
El modelo cliente-servidor es el que siguen la mayoría de aplicaciones de red. En este capítulo verás en qué consiste y cómo se pueden diseñar e implementar servidores eficientes y escalables aprovechando el soporte que ofrece la librería estándar de Python. También verás cómo aplicar técnicas de concurrencia, paralelismo y entrada/salida asíncrona.

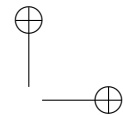
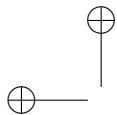
### **Publicador-subscriptor (C15)**

es otro de los modelos de interacción clásicos. Permite la comunicación entre múltiples participantes de forma más asíncrona y flexible que el modelo cliente-servidor, siempre claro, que el diseño de la aplicación lo permita. A pesar de no ser tan popular, es muy utilizado en aplicaciones de mensajería y propagación de notificaciones.

### **Serialización (C17)**

trata sobre cómo las aplicaciones de red pueden intercambiar datos arbitrariamente complejos en formato binario o texto de forma efi-





ciente. Verás las bases de la codificación de datos aplicada a código listo para probar.

### **Captura y análisis (C18)**

profundiza en el potencial que ofrecen las herramientas de captura de tráfico `tshark` y `wireshark`. El análisis del tráfico de red es importante para muchas situaciones y es clave en campos tan pujantes como la ciberseguridad. Aquí verás cómo capturar, filtrar y analizar tráfico de forma práctica y sencilla.

### **Sockets raw (C19)**

explica qué son y para qué se utilizan este tipo de sockets y su papel en el tratamiento y generación de mensajes a bajo nivel. Verás también ejemplos prácticos para escribir programas capaces de capturar y generar mensajes de red a cualquier nivel de la pila de protocolos.

FiXme  
Fatal!

### **FiXme Fatal: Completar resto de capítulos**

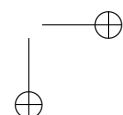
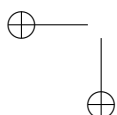
## **Anglicismos**

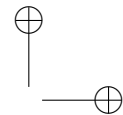
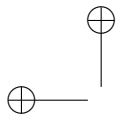
En el texto se utilizan muchos términos en inglés a pesar de que existen equivalentes en español. Esto es así porque priorizamos la claridad y la precisión técnica por encima de la corrección lingüística. Nuestro principal objetivo es que el lector entienda la redacción y comprenda los conceptos que se abordan, y eso no se consigue empleando términos que nadie utiliza en la ámbito profesional, por muy correctos que sean. Además, siempre que se introduzca un nuevo concepto, incluso si es de uso común en español, para evitar cualquier confusión se incluirá también el término en inglés entre paréntesis.

## **Hipertexto**

En la versión pdf del libro, todas las referencias a capítulos, secciones (denotadas con §), figuras, tablas, listados de código, citas bibliográficas y acrónimos son enlaces que te llevan al lugar dónde se encuentran. Pero en la mayoría de los casos basta con colocar el cursor del ratón encima del enlace para que el visor pdf muestre una vista previa.

En cuanto a los acrónimos, como en cualquier documento técnico, aparecen en gran cantidad. Cuando se trate de un acrónimo poco habitual se definirá la primera vez que se utilice. Si es de uso más común tendrás que consultar el listado de acrónimos en la pagina XVII.





## Ejemplos y código fuente

Todos los ejemplos que aparecen en el libro están disponibles para descarga a través del repositorio git en <https://github.com/destripando-internet/code>. Aunque es posible descargar este contenido individualmente o como un archivo comprimido, te aconsejamos utilizar el sistema de control de versiones git<sup>1</sup>. Las rutas relativas a este repositorio que aparecen en el texto son ‘clickables’ y van precedidas del logotipo de github (🐙).

Si encuentras alguna errata u omisión es los programas de ejemplo, por favor, utiliza la herramienta de gestión de incidencias (*issue tracker*) accesible desde <https://github.com/destripando-internet/code/issues> para notificarlo a sus autores.

## Sobre este documento

Este documento está tipografiado con L<sup>A</sup>T<sub>E</sub>X en un sistema Debian GNU/Linux. Las figuras y la mayoría de los diagramas están realizados con inkscape.

Los fuentes de este documento también se encuentran en un repositorio git en <https://github.com/destripando-internet/book>. Si quieres colaborar activamente en su desarrollo o mejora, ponte en contacto con los autores.

Al igual que con los ejemplos, también existe una herramienta pública de gestión de incidencias en la que puedes notificar problemas o errores de cualquier tipo que hayas detectado en el documento.

## Exención de responsabilidad

Este libro contiene abundantes ejemplos de comandos y programas destinados exclusivamente a fines formativos. En muchas ocasiones estos comandos implican privilegios de administración que pueden alterar la configuración y estado del sistema, y por tanto pueden tener consecuencias imprevistas y/o no deseables incluyendo la pérdida o exposición de datos o la interrupción de servicios.

Los autores y editores de este libro no son responsables de ningún daño o pérdida, directa o indirecta, como consecuencia de la utilización de los ejemplos o la información expuesta. Se recomienda al lector ejecutar siempre los ejemplos y pruebas en un entorno de laboratorio y nunca en sistemas en producción.

El lector asume toda responsabilidad por el uso que realice de los contenidos del libro.

---

<sup>1</sup><https://git-scm.com>

