La Guía Definitiva y Práctica de SSH y Certificados Digitales

De la Teoría a la Práctica en Windows, Linux y WSL

Introducción: Construyendo Confianza en un Mundo Digital

Cada día, realizamos acciones que dependen de una confianza invisible. ¿Cómo sabe tu navegador que cuando te conectas a la web de tu banco estás hablando realmente con el banco y no con un impostor? ¿Y cómo podemos, como administradores de sistemas, gestionar un servidor a miles de kilómetros de distancia de forma segura, sabiendo que nadie puede espiar lo que hacemos?

La respuesta a ambas preguntas se basa en una idea brillante y elegante: la **criptografía de clave pública**. Este es el cimiento sobre el que se construyen dos de las tecnologías de seguridad más importantes de la actualidad: los **Certificados Digitales** (la base de HTTPS) y el protocolo **Secure Shell (SSH)**.[1]

Esta guía es un viaje completo. Empezaremos por desmitificar los conceptos fundamentales y luego los aplicaremos en talleres prácticos paso a paso. El objetivo no es solo aprender a ejecutar comandos, sino entender profundamente *por qué* funcionan, para que puedas tomar decisiones informadas sobre la seguridad de tus sistemas.

Parte I: Los Cimientos - La Magia de la Criptografía de Clave Pública

Para construir una fortaleza digital, primero debemos entender sus materiales. La criptografía de clave pública es el ladrillo fundamental de la seguridad moderna.

1.1 El Par de Claves: Tu Identidad Digital Única

En el corazón de este sistema hay un par de claves matemáticamente vinculadas: una **clave pública** y una **clave privada**.[3] Se generan juntas, pero sus roles son opuestos y complementarios.

- **Clave Pública:** Como su nombre indica, está diseñada para ser compartida. Es tu "dirección de buzón" o el "dibujo de tu sello personal". Puedes dársela a cualquiera sin comprometer tu seguridad.
- **Clave Privada:** Es tu secreto más preciado. Es la "llave de tu buzón" o tu "sello de lacre personal". **Nunca, jamás, la compartes con nadie**.[3]

La magia reside en que lo que una clave cifra, solo la otra puede descifrarlo, y viceversa. Están diseñadas para que sea computacionalmente imposible adivinar la clave privada a partir de la pública.[3]

1.2 La Dualidad Clave: ¿Cifrar o Firmar?

Esta es la parte más brillante y la que más confunde al principio. El par de claves funciona en **ambas direcciones**, y cada dirección sirve para un propósito totalmente diferente.

Objetivo 1: Enviar un Secreto (CONFIDENCIALIDAD)

Quieres enviar un mensaje a un amigo y que **solo él** pueda leerlo.

1. **Tú (el emisor):** Coges la **clave pública de tu amigo** (su dirección de buzón).
2. Cifras el mensaje con su clave pública. Ahora el mensaje es ilegible para cualquiera que lo intercepte.
3. **Tu amigo (el receptor):** Recibe el mensaje. La única forma de descifrarlo es usando **su propia clave privada** (la llave de su buzón).

Regla de Oro para la Confidencialidad:
Para CIFRAR un mensaje para alguien, usas SU clave pública. Solo él podrá DESCIFRARLO con SU clave privada.6

Objetivo 2: Probar que eres Tú (AUTENTICACIÓN / FIRMA DIGITAL)

Ahora al revés. Quieres enviar un documento y que todo el mundo sepa sin lugar a dudas que lo has enviado tú y que no ha sido modificado.

1. **Tú (el emisor):** Usas **TU PROPIA clave privada** (tu sello de lacre personal) para crear una "firma digital" única para ese documento.
2. **El receptor:** Recibe el documento y la firma. Para comprobar si es auténtica, coge **TU clave pública** (el dibujo de tu sello que es público) y la usa para verificar la firma. Si la verificación funciona, sabe que el documento es tuyo y no ha sido alterado.[7]

Regla de Oro para la Autenticación:

Para FIRMAR un mensaje, usas TU clave privada. Cualquiera puede VERIFICAR esa firma usando TU clave pública.

Parte II: Certificados Digitales - El DNI de Internet

Si la criptografía de clave pública es el lenguaje de la confianza, los certificados digitales son los documentos oficiales que la formalizan. Son la razón por la que tu navegador muestra un candado al conectarse a tu banco.

2.1 ¿Qué es un Certificado Digital?

Un certificado digital es el **DNI o pasaporte digital de un sitio web** . Es un fichero que vincula una identidad (como www.ejemplo.com) con una clave pública. Crucialmente, este "DNI" está firmado digitalmente por una entidad de confianza, una **Autoridad de Certificación (CA)**.[9]

2.2 La Cadena de Confianza: ¿Por Qué Confías en tu Banco Online?

Tu navegador no confía ciegamente en el certificado que le da un sitio web. Confía en él porque puede verificar una **cadena de confianza** [9]:

1. **Certificados Raíz:** Tu sistema operativo y tu navegador vienen con una lista preinstalada de certificados de CAs raíz de máxima confianza (como DigiCert, GlobalSign, etc.).[12]
2. **Verificación en Cadena:** Cuando te conectas a https://mibanco.com, el servidor presenta su certificado. Tu navegador mira quién lo firmó.
3. Si el firmante es una de las CAs raíz de tu lista, la confianza se establece.
4. Si no, el navegador comprueba si el certificado del firmante fue, a su vez, firmado por otra CA (una CA intermedia), y así sucesivamente, hasta llegar a una CA raíz de confianza.[11]

Si se puede trazar una ruta válida hasta una CA raíz, el certificado es válido y la conexión es segura. Si no, el navegador muestra una advertencia.

2.3 Taller Práctico: El Certificado Autofirmado (¡Aprendiendo del Error!)

Vamos a ver este sistema en acción configurando un servidor web en **Ubuntu Server** con un certificado que firmaremos nosotros mismos. Esto nos permitirá entender por qué el

navegador se queja.

Paso 1: Instalar Nginx en Ubuntu Server

Primero, necesitamos un servidor web.

Bash

```
sudo apt update
sudo apt install nginx -y
sudo systemctl status nginx
```

Paso 2: Generar Clave Privada y Certificado Autofirmado

Ahora, crearemos nuestro propio "DNI" digital. Como lo firmamos nosotros y no una CA oficial, se llama "autofirmado".

Bash

```
# Creamos los directorios para guardar la clave y el certificado
sudo mkdir -p /etc/ssl/private
sudo mkdir -p /etc/ssl/certs

# Generamos la clave y el certificado en un solo paso
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/nginx-selfsigned.key \
-out /etc/ssl/certs/nginx-selfsigned.crt
```

Desglose del comando [2]:

- openssl req -x509: Herramienta para crear un certificado X.509.
- -nodes: No cifra la clave privada con una contraseña (para que el servidor reinicie sin intervención).
- -days 365: Validez de un año.
- -newkey rsa:2048: Genera una nueva clave privada RSA de 2048 bits.
- -keyout y -out: Rutas para guardar la clave privada y el certificado.

El comando te hará preguntas. La más importante es Common Name. Aquí debes poner la **dirección IP de tu servidor Ubuntu**.[13]

Paso 3: Configurar Nginx para Usar el Certificado (HTTPS)

Le diremos a Nginx que use nuestros nuevos ficheros para conexiones seguras.

Bash

```
# Creamos un nuevo fichero de configuración para nuestro sitio seguro
sudo nano /etc/nginx/sites-available/mi_sitio_seguro
```

Pega la siguiente configuración, **reemplazando ip_del_servidor** con la IP real de tu VM:

Nginx

```
server {
    listen 443 ssl;
    listen [::]:443 ssl;

    server_name ip_del_servidor;

    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

    root /var/www/html;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Ahora, activamos el sitio, probamos la sintaxis y reiniciamos Nginx:

Bash

```
sudo ln -s /etc/nginx/sites-available/mi_sitio_seguro /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

Paso 4: Ajustar el Firewall y Probar

Permitimos el tráfico en el puerto 443 (HTTPS) a través del firewall ufw:

Bash

```
sudo ufw allow 'Nginx Full'
sudo ufw status
```

Ahora, desde un navegador en tu máquina anfitriona (Windows o Linux), accede a tu servidor usando https://ip_del_servidor.

El Momento de la Verdad

Tu navegador mostrará una advertencia: **"Su conexión no es privada"**.[2] ¡Esto es perfecto! Es el punto de aprendizaje clave. La advertencia no significa que la conexión no esté cifrada; de hecho, lo está. Lo que el navegador te dice es: "He recibido un DNI, pero el emisor de este DNI es el propio titular. No puedo verificar su identidad con ninguna de las autoridades en las que confío".

Parte III: Secure Shell (SSH) - La Llave Maestra para tus Servidores

Si los certificados son el pasaporte para el público, SSH es la llave maestra de alta seguridad para los administradores.

3.1 El Duelo: Contraseñas vs. Claves SSH

SSH admite dos métodos de autenticación: contraseñas y claves. Aunque las contraseñas son familiares, son mucho menos seguras.[14]

- **Autenticación por Contraseña:** Se basa en "algo que sabes". Es vulnerable a ataques de fuerza bruta (bots probando miles de contraseñas por segundo) y contraseñas débiles.[14]
- **Autenticación por Clave SSH:** Se basa en "algo que tienes" (tu clave privada). El servidor te envía un desafío cifrado con tu clave pública, y solo tu clave privada puede

resolverlo correctamente. La clave privada nunca viaja por la red.[17]

3.2 El Ecosistema de Ficheros ~/.ssh

Tanto en tu máquina local (cliente) como en el servidor, el directorio ~/.ssh es el centro de operaciones.

Fichero	¿Dónde está?	¿Qué es?	¿Para qué sirve?
id_rsa o id_ed25519	**Tu PC (Cliente)**	Tu **CLAVE PRIVADA**.	¡Tu identidad! Es la llave. **¡NUNCA LA COMPARTAS!** Sus permisos deben ser restrictivos .
id_rsa.pub o id_ed25519.pub	**Tu PC (Cliente)**	Tu **CLAVE PÚBLICA**.	El "candado" o el "dibujo de tu sello". Este es el fichero que copias a los servidores a los que quieres acceder .
authorized_keys	**Servidor Remoto**	Una lista de claves públicas.	Es el "llavero" del servidor. Contiene las claves públicas de todos los usuarios que tienen permiso para entrar .
known_hosts	**Tu PC (Cliente)**	Una lista de las claves públicas de los servidores.	Es tu "agenda de confianza". Guarda la identidad de cada servidor al que te conectas para protegerte de ataques de suplantación .

Parte IV: Taller Práctico Integral de SSH (Windows, WSL y Ubuntu)

Ahora aplicaremos toda la teoría. El flujo de trabajo es siempre el mismo: generar claves en nuestra máquina local, copiar la clave pública al servidor y conectarnos.

4.1 Generación de Claves en tu Máquina Local

El primer paso es crear tu par de claves personal.

Práctica en Windows (usando PowerShell)

Las versiones modernas de Windows 10 y 11 incluyen un cliente OpenSSH nativo.[20]

1. Abre **PowerShell**.
2. Ejecuta el comando ssh-keygen. Recomendamos usar el algoritmo Ed25519 por ser más moderno y seguro.[22]
   PowerShell
   ssh-keygen -t ed25519

3. El programa te hará preguntas:
   - Enter file in which to save the key...: Pulsa **Enter** para aceptar la ruta por defecto (C:\Users\<TuUsuario>\.ssh\id_ed25519).[5]
   - Enter passphrase...: Puedes introducir una contraseña para proteger tu clave privada en tu disco. Es una capa extra de seguridad. Para empezar, puedes dejarla en blanco pulsando **Enter** dos veces.[18]

Práctica en Linux (Ubuntu Server) y WSL

El proceso dentro de una terminal de Linux o WSL es idéntico.

1. Abre una terminal de Ubuntu o WSL.
2. Ejecuta el mismo comando ssh-keygen:
   Bash
   ssh-keygen -t ed25519

3. Las claves se guardarán en la ruta ~/.ssh/ (que es /home/<tu_usuario>/.ssh/).[4]

4.2 Concediendo Acceso al Servidor Ubuntu

Ahora tenemos que poner nuestra clave pública (id_ed25519.pub) en el fichero authorized_keys del servidor.

Práctica desde WSL o Linux (Método Recomendado)

La herramienta ssh-copy-id es la forma más sencilla y segura, ya que automatiza el proceso y se asegura de que los permisos sean correctos.[1]

1. Desde tu terminal de WSL o Linux, ejecuta:
   Bash
   # Sustituye 'usuario' y 'ip_del_servidor' por los datos reales
   ssh-copy-id usuario@ip_del_servidor

2. Te pedirá la contraseña del usuario en el servidor remoto. **Esta será la última vez que la uses**.[28]

Práctica desde Windows PowerShell (Método Manual)

PowerShell no tiene ssh-copy-id, así que lo haremos a mano para entender lo que pasa por debajo.

1. Desde PowerShell, ejecuta el siguiente comando en una sola línea. Este comando lee tu clave pública y la añade al fichero authorized_keys en el servidor.
   PowerShell
   # Reemplaza 'usuario' e 'ip_del_servidor'
   cat $env:USERPROFILE\.ssh\id_ed25519.pub | ssh usuario@ip_del_servidor "mkdir -p ~/.ssh; cat >> ~/.ssh/authorized_keys; chmod 700 ~/.ssh; chmod 600 ~/.ssh/authorized_keys"

   Desglose del comando [28]:
   - cat...: Lee el contenido de tu clave pública.
   - | ssh...: Se conecta al servidor y ejecuta los comandos entre comillas.
   - mkdir...; chmod...: Crea el directorio .ssh si no existe y le da los permisos correctos (muy importante).
   - cat >>...; chmod...: Añade tu clave al fichero authorized_keys y le da los permisos correctos.
2. Te pedirá la contraseña del servidor por última vez.

4.3 La Conexión Segura y el Fichero known_hosts

Ahora, intenta conectarte desde PowerShell, WSL o Linux:

Bash

ssh usuario@ip_del_servidor

La primera vez que te conectes, verás un mensaje como este:

The authenticity of host '...' can't be established.
Are you sure you want to continue connecting (yes/no)?

Este es un paso de seguridad vital. Tu cliente te está mostrando la "huella digital" del servidor y te pregunta si confías en él. Escribe yes y pulsa Enter. Al hacerlo, tu cliente guarda la clave pública del servidor en tu fichero local ~/.ssh/known_hosts. En futuras conexiones, si la clave del servidor no coincide, te lanzará una advertencia grave, protegiéndote de ataques de suplantación .
¡Deberías entrar directamente sin que te pida contraseña!

4.4 Fortaleciendo el Servidor: Deshabilitar Contraseñas

Ahora que podemos entrar con nuestra llave segura, vamos a tapiar la puerta de atrás para que los atacantes no puedan intentar adivinar contraseñas.

1. Conéctate a tu **servidor Ubuntu** por SSH.
2. Edita el fichero de configuración del servicio SSH con nano:
   Bash
   sudo nano /etc/ssh/sshd_config

3. Busca la línea que dice #PasswordAuthentication yes o PasswordAuthentication yes.
4. Quítale la # si la tiene y cambia yes por no [29]:
   PasswordAuthentication no

5. Guarda el fichero (Ctrl+O en nano) y sal (Ctrl+X).
6. Reinicia el servicio SSH para que aplique los cambios:
   Bash
   sudo systemctl restart sshd

¡Listo! Ahora tu servidor solo acepta conexiones con las claves SSH que tú autorices. Has eliminado el vector de ataque más común.[31]

Parte V: Profundizando - Las Capas de Red (Modelo OSI)

Para entender la diferencia entre herramientas como una VPN y un túnel SSH, es útil conocer

las "capas" de comunicación de una red.

5.1 La Analogía del Servicio Postal

Imagina que la comunicación en red es como enviar una carta:

- **Capa 7 (Aplicación):** Es el **contenido de la carta** (el texto, las fotos). Aquí viven protocolos como HTTP (web), SMTP (email) y **SSH** .
- **Capa 4 (Transporte):** Es el **tipo de envío**. Puedes elegir un envío certificado y fiable (TCP) o una postal rápida y sin garantías (UDP) .
- **Capa 3 (Red):** Es la **dirección en el sobre** (la dirección IP). Aquí trabajan los routers para decidir la mejor ruta .

5.2 ¿Qué significa "trabajar en una capa"?

- **Túnel SSH (Nivel de Aplicación - Capa 7):** Es como un **mensajero personal** al que le das una instrucción muy específica: "Coge los paquetes del buzón 5432, mételos en un maletín blindado y llévalos a la habitación 5432 del destino". Solo funciona para ese servicio concreto .
- **VPN (Nivel de Red - Capa 3):** Es como construir un **túnel subterráneo privado** entre dos casas. Cualquier vehículo (TCP, UDP, ping) que quiera ir a la otra casa usará ese túnel. Es transparente para las aplicaciones y da acceso a toda la red .

5.3 La Misteriosa Capa de Sesión (Capa 5)

Esta capa es el **"moderador de la conversación"**. Se encarga de iniciar, mantener y terminar un diálogo entre dos aplicaciones. En el mundo real de Internet, sus funciones suelen estar integradas en los propios protocolos de aplicación. La ves en acción en:

- **Videollamadas:** Mantiene el audio y el vídeo sincronizados como parte de la misma "sesión".
- **Juegos Online:** Mantiene tu conexión y estado de juego con el servidor.
- **Descargas grandes:** Permite reanudar una descarga interrumpida desde un punto de control.

Conclusión

Hemos recorrido un largo camino, desde la teoría de la criptografía hasta la configuración práctica y segura de servidores. Ahora sabes que, aunque los Certificados Digitales y SSH sirven para propósitos diferentes, ambos se basan en el mismo principio de clave pública. La

diferencia clave radica en su modelo de confianza:

- **Certificados Digitales:** Confianza **centralizada y jerárquica** (PKI), para que el público general pueda confiar en sitios web.
- **SSH:** Confianza **descentralizada y explícita**, para que los administradores gestionen sistemas de forma segura.

Con estas herramientas y conocimientos, tienes el poder de construir y mantener sistemas más seguros y robustos.

Obras citadas

1. Criptografía asimétrica | Cloud Key Management Service, fecha de acceso: octubre 14, 2025, https://cloud.google.com/kms/docs/asymmetric-encryption?hl=es-419
2. ¿Qué es SSH? | Protocolo Secure Shell (SSH) - Cloudflare, fecha de acceso: octubre 14, 2025, https://www.cloudflare.com/es-es/learning/access-management/what-is-ssh/
3. Criptografía de clave asimétrica - CERES - FNMT, fecha de acceso: octubre 14, 2025, https://www.cert.fnmt.es/curso-de-criptografia/criptografia-de-clave-asimetrica
4. ¿Qué es el cifrado asimétrico? - IBM, fecha de acceso: octubre 14, 2025, https://www.ibm.com/es-es/think/topics/asymmetric-encryption
5. Qué es una clave SSH: autenticación segura sin contraseña - Raiola Networks, fecha de acceso: octubre 14, 2025, https://raiolanetworks.com/ayuda/clave-ssh/
6. vpn vs ssh, what to choose? [duplicate] - Information Security Stack Exchange, fecha de acceso: octubre 16, 2025, https://security.stackexchange.com/questions/109248/vpn-vs-ssh-what-to-choose
7. Generate SSH keys en Ubuntu: crear y configurar una clave SSH - IONOS, fecha de acceso: octubre 14, 2025, https://www.ionos.com/es-us/digitalguide/servidores/configuracion/generar-claves-ssh-en-ubuntu/
8. Guía para configurar claves SSH en un servidor Linux [Tutorial] - Stackscale, fecha de acceso: octubre 14, 2025, https://www.stackscale.com/es/blog/configurar-llaves-ssh-servidor-linux/
9. Proteger su servidor SSH - IBM, fecha de acceso: octubre 14, 2025, https://www.ibm.com/docs/es/aspera-fasp-proxy/1.4.x?topic=appendices-securing-your-ssh-server
10. Cómo configurar SSH sin contraseña en Linux - Hostinger, fecha de acceso: octubre 14, 2025, https://www.hostinger.com/es/tutoriales/configurar-ssh-sin-contrasena-linux
11. VPN vs SSH tunneling: Key differences in secure remote access, fecha de acceso: octubre 16, 2025, https://eureka.patsnap.com/article/vpn-vs-ssh-tunneling-key-differences-in-secure-remote-access
12. Claves SSH entre Windows 10 y Linux : r/linuxadmin - Reddit, fecha de acceso:

octubre 14, 2025, https://www.reddit.com/r/linuxadmin/comments/1j0ziba/ssh_keys_between_windows_10_and_linux/?tl=es-419
13. ¿Qué son los certificados digitales? | Fortinet, fecha de acceso: octubre 14, 2025, https://www.fortinet.com/lat/resources/cyberglossary/digital-certificates
14. Llaves SSH vs contraseña : r/linuxquestions - Reddit, fecha de acceso: octubre 14, 2025, https://www.reddit.com/r/linuxquestions/comments/ehapud/ssh_keys_vs_password/?tl=es-419
15. What significance does the user/host at the end of an SSH public key file hold?, fecha de acceso: octubre 14, 2025, https://serverfault.com/questions/743548/what-significance-does-the-user-host-at-the-end-of-an-ssh-public-key-file-hold
16. Explicación de las claves SSH: Guía para un acceso remoto rápido y seguro - DataCamp, fecha de acceso: octubre 14, 2025, https://www.datacamp.com/es/tutorial/ssh-keys
17. Criptografía asimétrica - Wikipedia, la enciclopedia libre, fecha de acceso: octubre 14, 2025, https://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica
18. Usando OpenSSL para generar y formatear certificados - IBM, fecha de acceso: octubre 14, 2025, https://www.ibm.com/docs/es/api-connect/10.0.x_cd?topic=profile-using-openssl-generate-format-certificates
19. Configura clave SSH en GitHub para Windows 11 con Power-Shell - Medium, fecha de acceso: octubre 14, 2025, https://medium.com/@aitorevi/configura-clave-ssh-en-github-para-windows-11-con-power-shell-afb51f818328
20. Cómo crear un certificado SSL autofirmado para Apache en Ubuntu 18.04 - DigitalOcean, fecha de acceso: octubre 14, 2025, https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-18-04-es
21. Add public key to known_hosts file - linux - Stack Overflow, fecha de acceso: octubre 14, 2025, https://stackoverflow.com/questions/34906302/add-public-key-to-known-hosts-file
22. SSH: qué es y cómo funciona este protocolo | Blog de Arsys, fecha de acceso: octubre 14, 2025, https://www.arsys.es/blog/ssh
23. can I share my SSH keys between WSL and Windows? - Super User, fecha de acceso: octubre 14, 2025, https://superuser.com/questions/1183176/can-i-share-my-ssh-keys-between-wsl-and-windows
24. Do I need to name my public RSA key to "authorized_keys" in ~/.ssh folder? - Reddit, fecha de acceso: octubre 14, 2025, https://www.reddit.com/r/linuxquestions/comments/uipm15/do_i_need_to_name_my_public_rsa_key_to_authorized/

25. www.lenovo.com, fecha de acceso: octubre 14, 2025, https://www.lenovo.com/es/es/glossary/digital-certificate/#:~:text=Un%20certificado%20digital%20es%20un,de%20confianza%20entre%20las%20partes.
26. Como generar una clave SSH en Windows - Tecnocratica, fecha de acceso: octubre 14, 2025, https://tecnocratica.net/wikicratica/books/windows/page/como-generar-una-clave-ssh-en-windows
27. Why VPN when I can just use SSH keys? - Server Fault, fecha de acceso: octubre 16, 2025, https://serverfault.com/questions/348191/why-vpn-when-i-can-just-use-ssh-keys
28. ¿Cuál es la diferencia entre claves PGP, claves SSH y claves SSL? - Quora, fecha de acceso: octubre 14, 2025, https://es.quora.com/Cu%C3%A1l-es-la-diferencia-entre-claves-PGP-claves-SSH-y-claves-SSL
29. SSH: Guía básica para aprender a utilizar este protocolo - Raiola Networks, fecha de acceso: octubre 14, 2025, https://raiolanetworks.com/blog/ssh/
30. SSH Copy ID for Copying SSH Keys to Servers, fecha de acceso: octubre 14, 2025, https://www.ssh.com/academy/ssh/copy-id
31. Certificados digitales - IBM, fecha de acceso: octubre 14, 2025, https://www.ibm.com/docs/es/integration-bus/10.0.0?topic=overview-digital-certificates
32. How to Set Up SSH Keys on Ubuntu: A Comprehensive Guide - DigitalOcean, fecha de acceso: octubre 14, 2025, https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-ubuntu-22-04
33. Is there an equivalent to ssh-copy-id for Windows? - Server Fault, fecha de acceso: octubre 14, 2025, https://serverfault.com/questions/224810/is-there-an-equivalent-to-ssh-copy-id-for-windows