

# EXPOSICION 2

## Equipo

ESCOM  
8

Equipo:

- Ramirez Contreras Angel Humberto
- Reyes Vivar Fernando
- Morales Torres Alejandro

Temas:

- Aplicaciones no orientadas a conexión 4.1
- Protocolo FTP 4.1.1
- Protocolo DNS 4.1.2



# CONTENIDO

## Introduccion

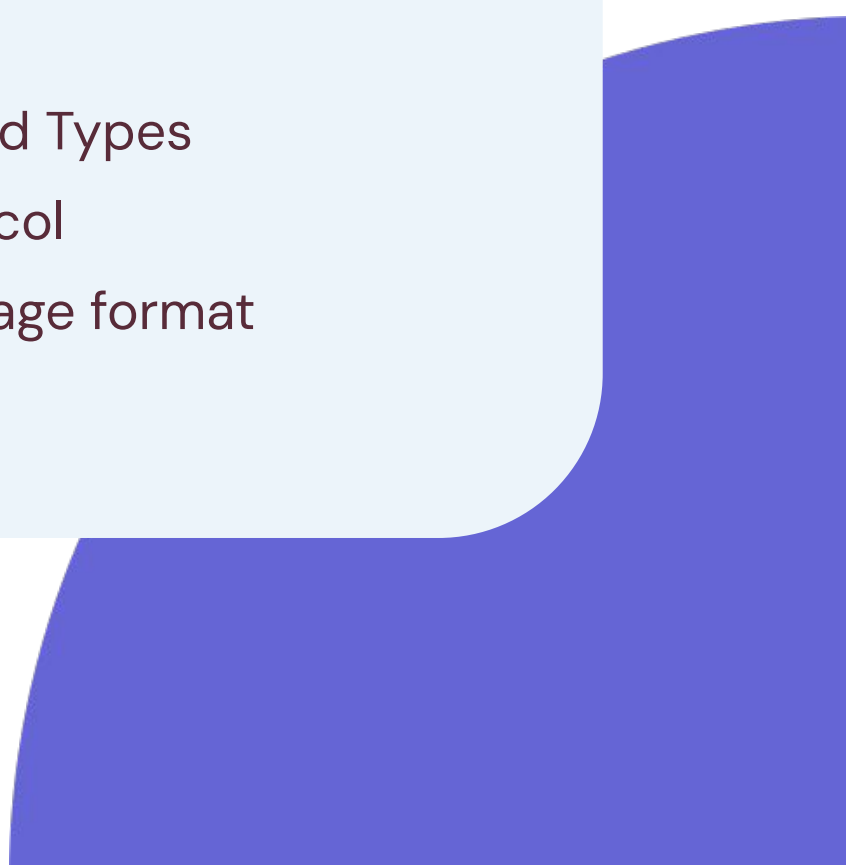
### Aplicaciones no orientadas a conexion

- Retomando sockets
- Tipos de sockets
- Protocolos
- Modulo de red
- Aplicación
- Programa UDP

## Protocolo FTP

- FTP
- ¿Dónde se utiliza?
- Paradigma cliente-servidor
- Archivos en un servidor FTP remoto
- Subir un archivo local a un servidor FTP remoto
- Copiar un archivo a una máquina remota por SFTP

## Protocolo DNS

- Hostname Resolution y DNS
  - ¿Qué es el protocolo DNS?
  - Resolucion de un nombre de dominio
  - DNS Record Types
  - DNS protocol
  - DNS message format
- 

# INTRODUCCION

---

ESTA EXPOSICIÓN EXAMINA LAS APLICACIONES NO ORIENTADAS A CONEXION, EL PROTOCOLO DE TRANSFERENCIA DE ARCHIVOS (FTP) Y EL SISTEMA DE NOMBRES DE DOMINIO (DNS), DESTACANDO SU PAPEL ESENCIAL EN LA TRANSFERENCIA DE ARCHIVOS Y LA RESOLUCIÓN DE NOMBRES EN INTERNET.





# **Aplicaciones no orientadas a conexión**

**Ramirez Contreras Angel  
Humberto**



**2**

# Retomando sockets

Para entender las aplicaciones no orientas a conexión, debemos retomar Sockets, de manera rápida, un socket es una manera de comunicarse con otros programas utilizando un file descriptor.

# TIPOS DE SOCKETS

EXISTEN TODO TIPO DE SOCKETS. ESTÁN LAS DIRECCIONES DE INTERNET DARPA (SOCKETS DE INTERNET), NOMBRES DE RUTA EN UN NODO LOCAL (SOCKETS UNIX), DIRECCIONES CCITT X.25 Y PROBABLEMENTE MUCHOS OTROS DEPENDIENDO DEL SABOR DE UNIX QUE USES



# PROTOSCOLOS

EL PROTOCOLO TFTP DICE QUE POR CADA PAQUETE QUE SE ENVÍA, EL DESTINATARIO TIENE QUE ENVIAR DE VUELTA UN PAQUETE QUE DICE: "¡LO RECIBÍ!" (UN PAQUETE "ACK"). SI EL EMISOR DEL PAQUETE ORIGINAL NO RECIBE UNA RESPUESTA, DIGAMOS, EN CINCO SEGUNDOS, RETRANSMITIRÁ EL PAQUETE HASTA QUE FINALMENTE RECIBA UN ACK.



# MODELO DE RED

LA CAPA FÍSICA ES EL HARDWARE. LA CAPA DE APLICACIÓN ESTÁ TAN LEJOS DE LA CAPA FÍSICA COMO PUEDAS IMAGINAR: ES EL LUGAR DONDE LOS USUARIOS INTERACTÚAN CON LA RED.

AHORA, ESTE MODELO ES TAN GENERAL QUE PROBABLEMENTE PODRÍAS USARLO COMO UNA GUÍA-

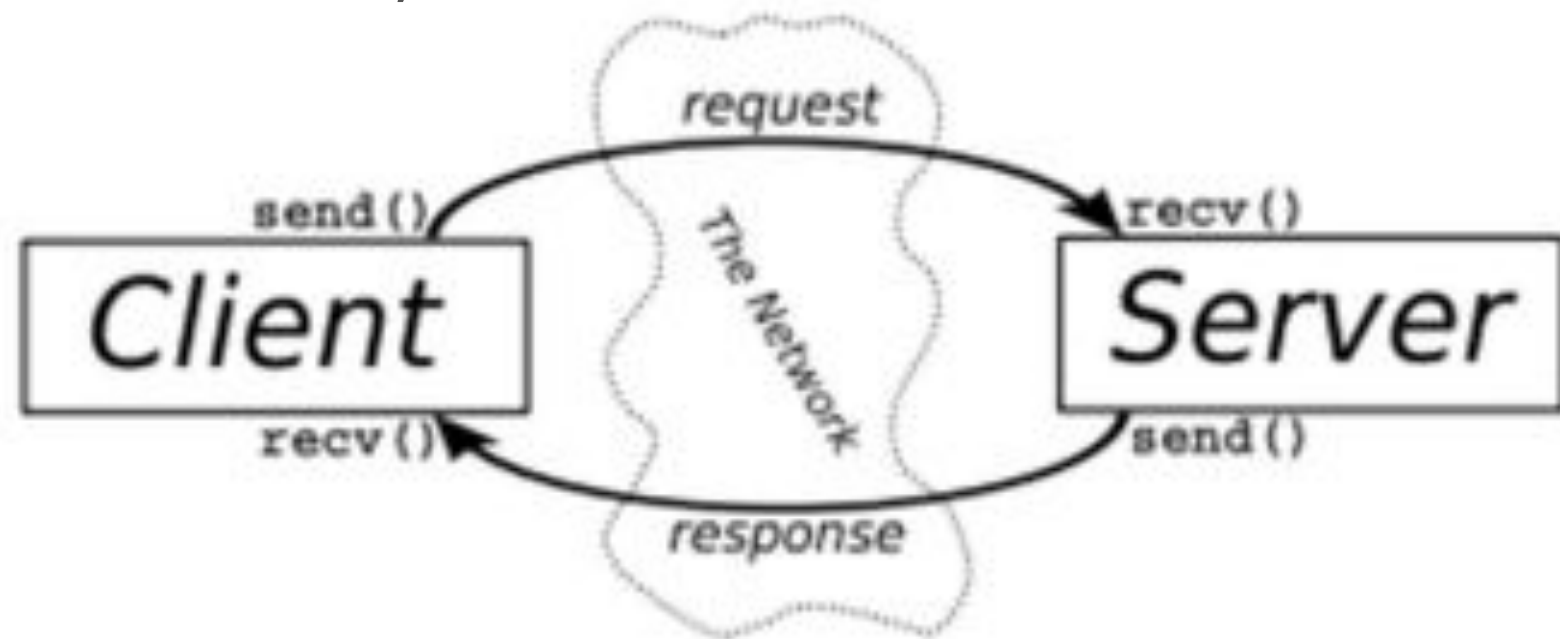
- CAPA DE APLICACIÓN (TELNET, FTP, ETC.)
- CAPA DE TRANSPORTE DE HOST A HOST (TCP, UDP)



# APLICACIÓN

Un ejemplo de una aplicación orientada a no conexión puede ocupar telnet, debemos conectarnos a un host remoto en el puerto 23 (el cliente), un programa en ese host (llamado telnet, el servidor) se activa.

Maneja la conexión entrante de telnet, te proporciona un prompt de inicio de sesión, etc.

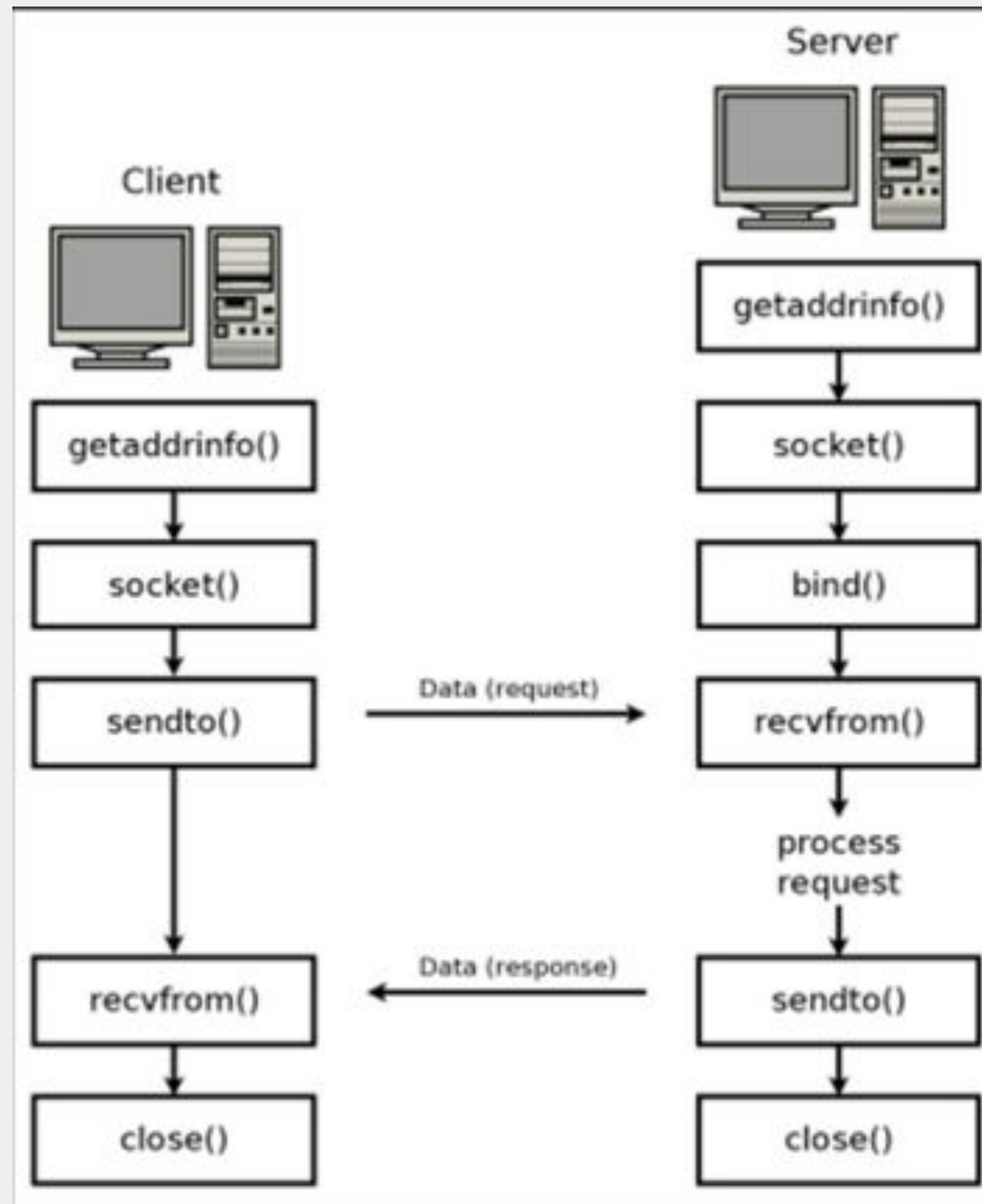


# PROGRAMA UDP

Un cliente UDP debe conocer la dirección del par UDP remoto para enviar el primer paquete.

Es importante saber que el cliente UDP no puede recibir datos primero, ya que el par remoto no tendría forma de saber a dónde enviar los datos sin recibir primero datos del cliente.

Esto es diferente de TCP, donde primero se establece una conexión con un saludo inicial. En TCP, tanto el cliente como el servidor pueden enviar los primeros datos de aplicación. Un servidor UDP escucha conexiones de un cliente UDP.



Flujo de programa UDP

Reyes Vivar Fernando

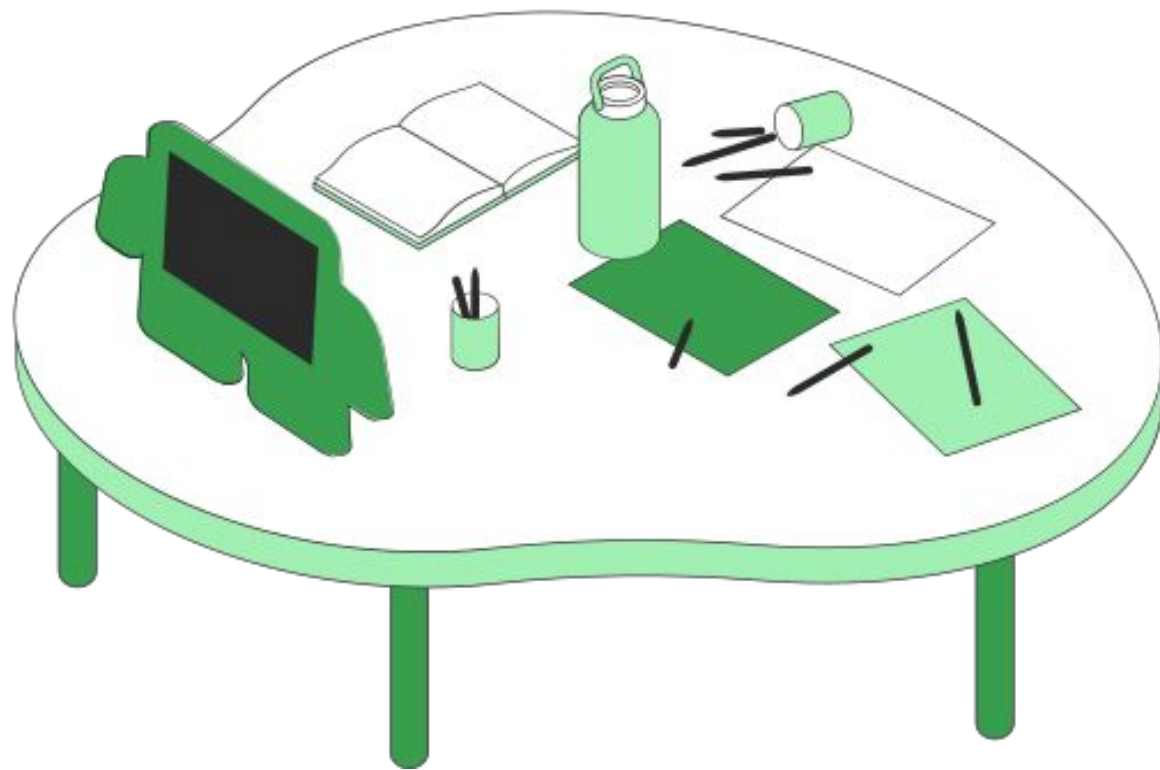
# PROTOCOLO FTP



10

# Ftp

## Indice



---

Dónde se utiliza

---

---

Paradigma cliente-servidor

---

---

Archivos en un servidor FTP remoto

---

---

Subir un archivo local a un servidor FTP remoto

---

---

Copiar un archivo a una máquina remota por SFTP

---

# ¿Dónde se utiliza?

TCP es utilizado por muchos protocolos. HTTP (para servir páginas web), FTP (para transferir archivos), SSH (para administración remota) y SMTP (para correo electrónico) todos usan TCP





A photograph of a workspace on the left side of the slide. It features a white desk with a gold-colored laptop, a small potted plant with green leaves, and a white ceramic mug. A blue and white patterned object is visible in the bottom left corner.

# Paradigma cliente -servidor

El servidor FTP escucha conexiones hasta que el cliente FTP se conecta. Después de la conexión inicial, el cliente FTP emite comandos al servidor. Si el cliente FTP solicita un archivo del servidor, el servidor intentará establecer una nueva conexión con el cliente FTP para transferir el archivo



# Archivos en un servidor FTP remoto

Una sesión de cliente FTP se puede crear llamando a la función `FTP()` de `ftplib` con las credenciales de la conexión anterior. Esto devuelve un manejador de cliente, que luego puede ser usado para ejecutar los comandos FTP habituales, como el comando para cambiar el directorio de trabajo o `cwd()`

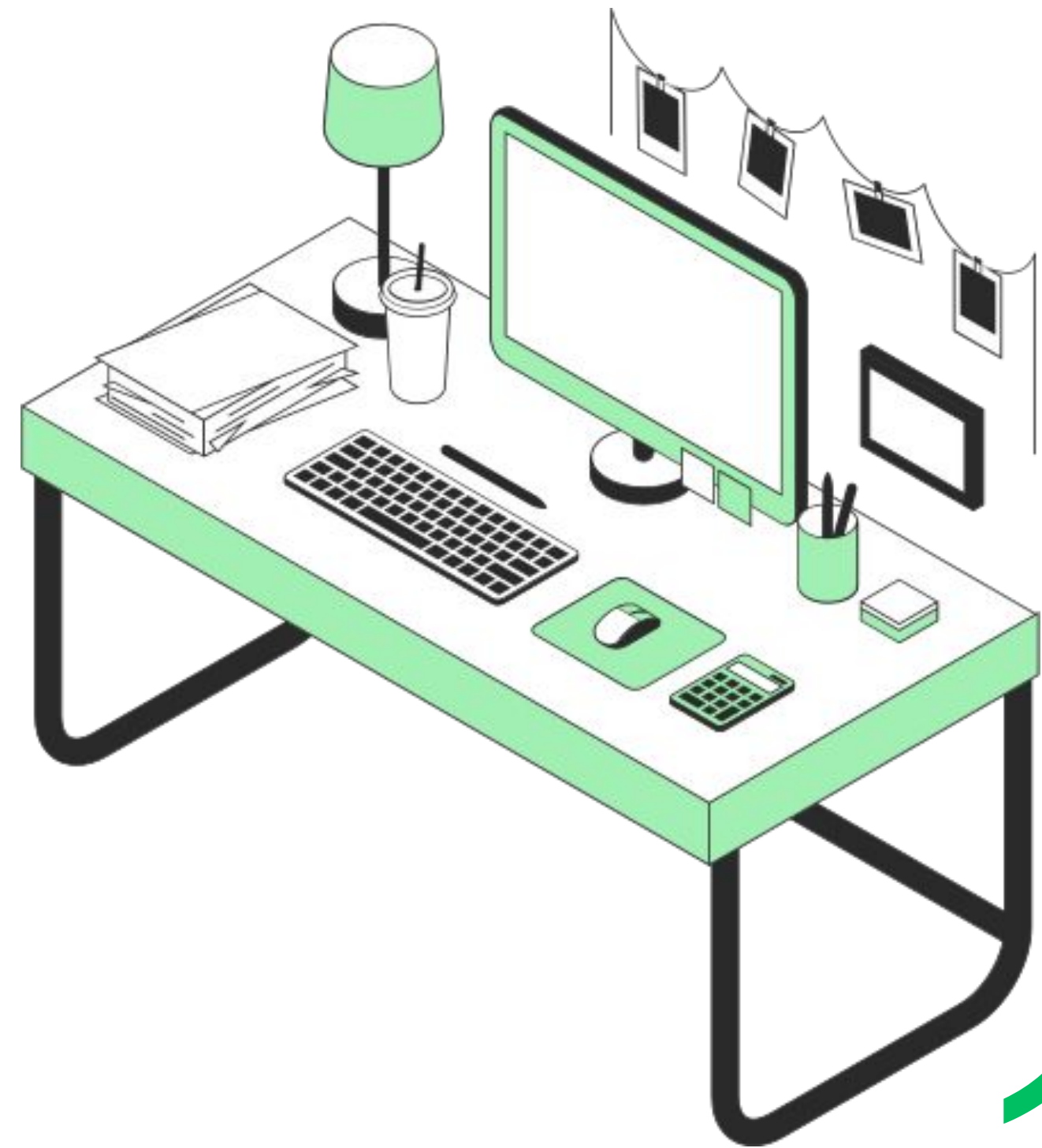


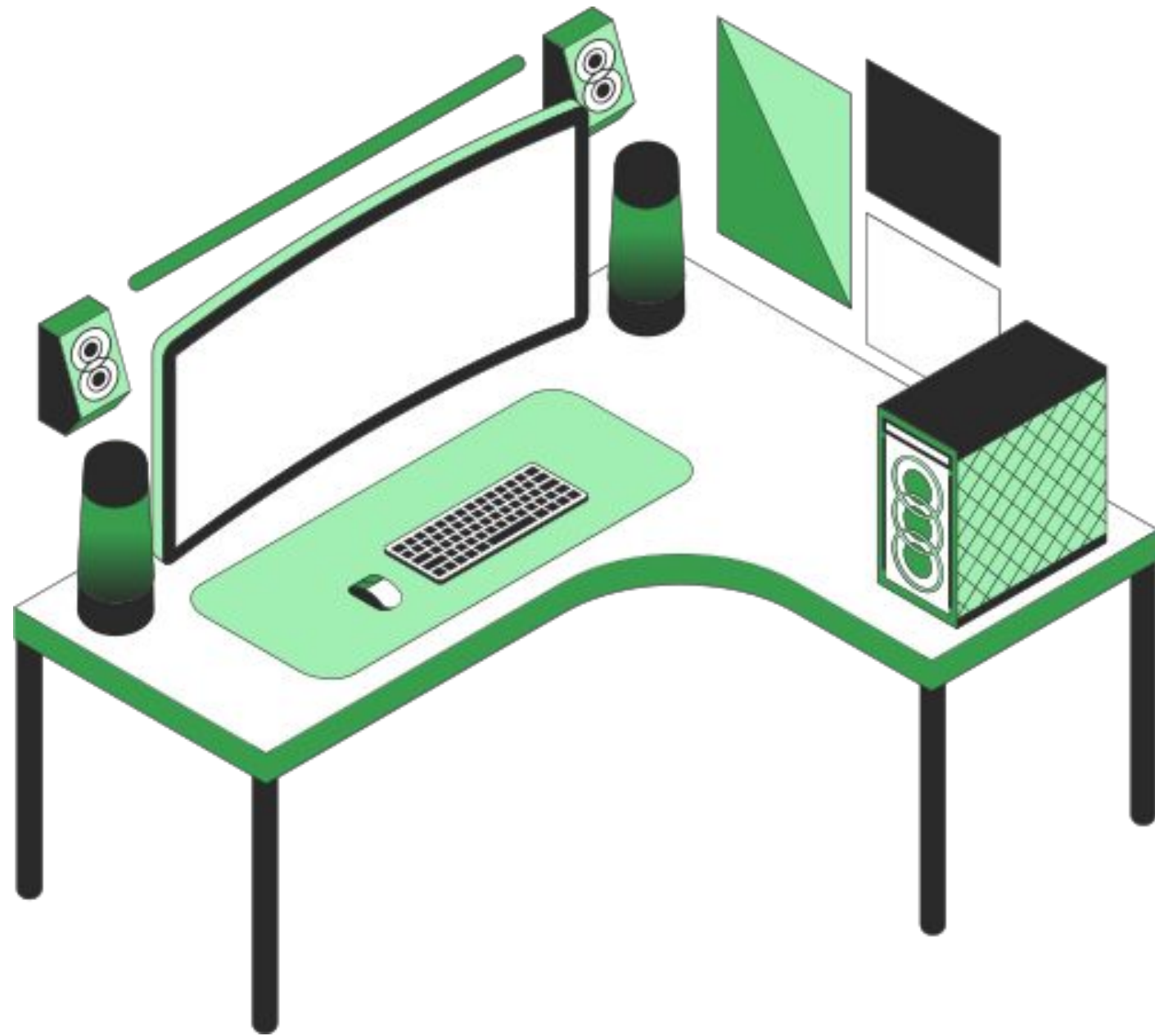


# Subir un archivo local a un servidor FTP remoto

En esta receta, asumimos que un servidor FTP local está en funcionamiento.

Alternativamente, puedes conectarte a un servidor FTP remoto. El método `ftp_upload()` utiliza la función `FTP()` de `ftplib` de Python para crear un objeto de conexión FT





# Copiar un archivo a una máquina remota por SFTP

Esta receta puede tomar varias entradas para conectarse a una máquina remota y copiar un

archivo por SFTP. Esta receta pasa la entrada de línea de comando a la función `copy_file()`.

Luego crea un cliente SSH llamando a la clase `SSHClient` de `paramiko`.

1



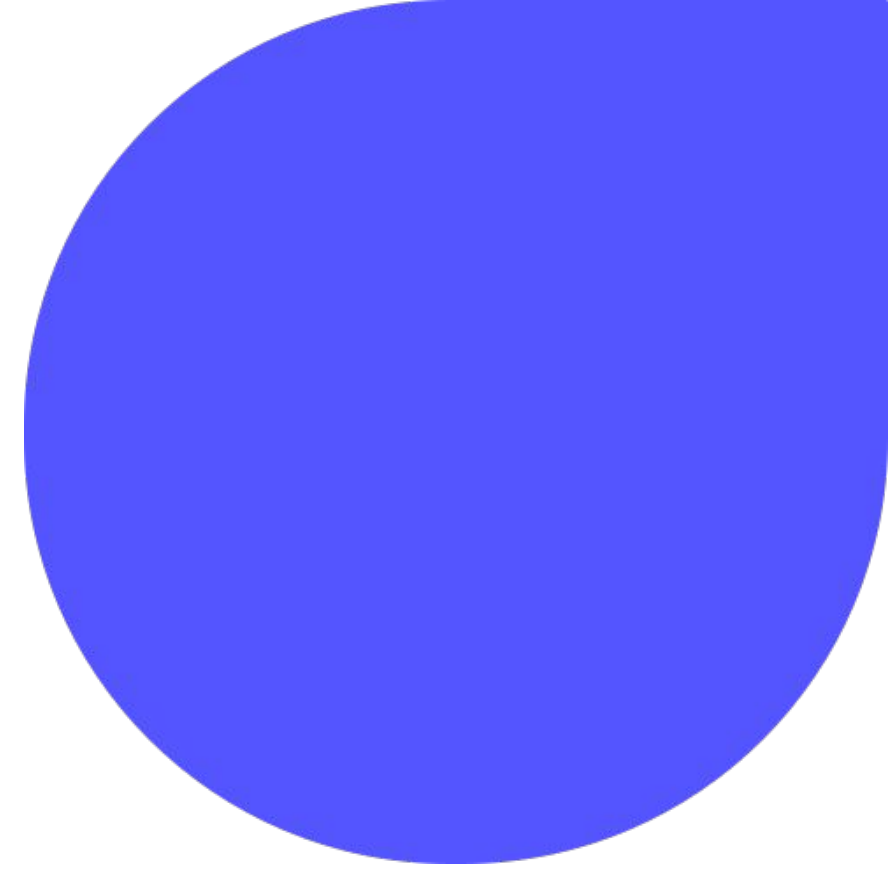
# PROTOCOLO DNS

Morales Torres Alejandro



17

# HOSTNAME RESOLUTION Y DNS



La resolución de nombres de dominio nos permite usar nombres tal como [www.facebook.com](http://www.facebook.com) en vez de la dirección propia del sitio como lo puede ser `::ffff:192.168.212.115` y precisamente el mecanismo que hace posible resolver el nombre de host en una dirección IP es DNS

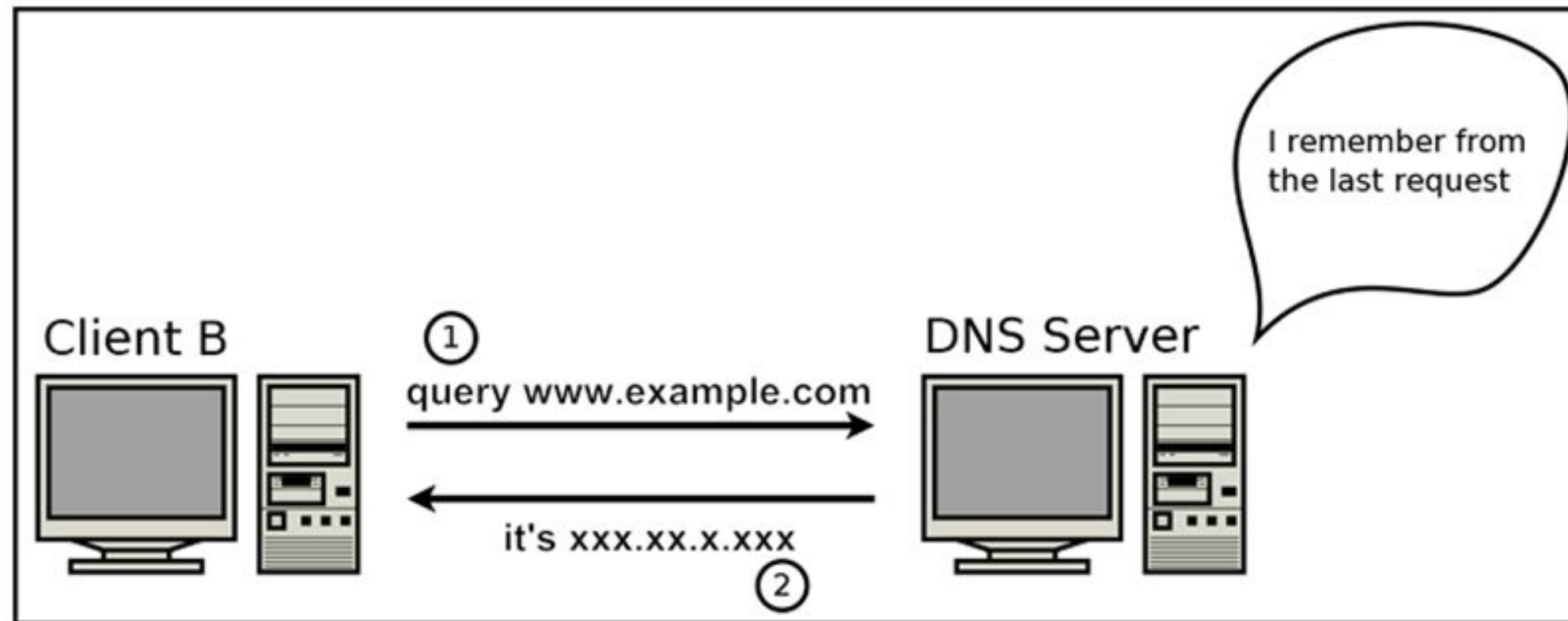


# ¿Qué es el protocolo DNS?

DNS significa Domain Name Service (servicio de nombre de dominio) y funciona de tal forma que asigna nombres a las computadoras y sistemas conectadas a internet.

Funciona gracias a la llamada de la función `getaddrinfo()` y el sistema operativo después se encarga de hacer lo necesario para la resolución de ese nombre de dominio.

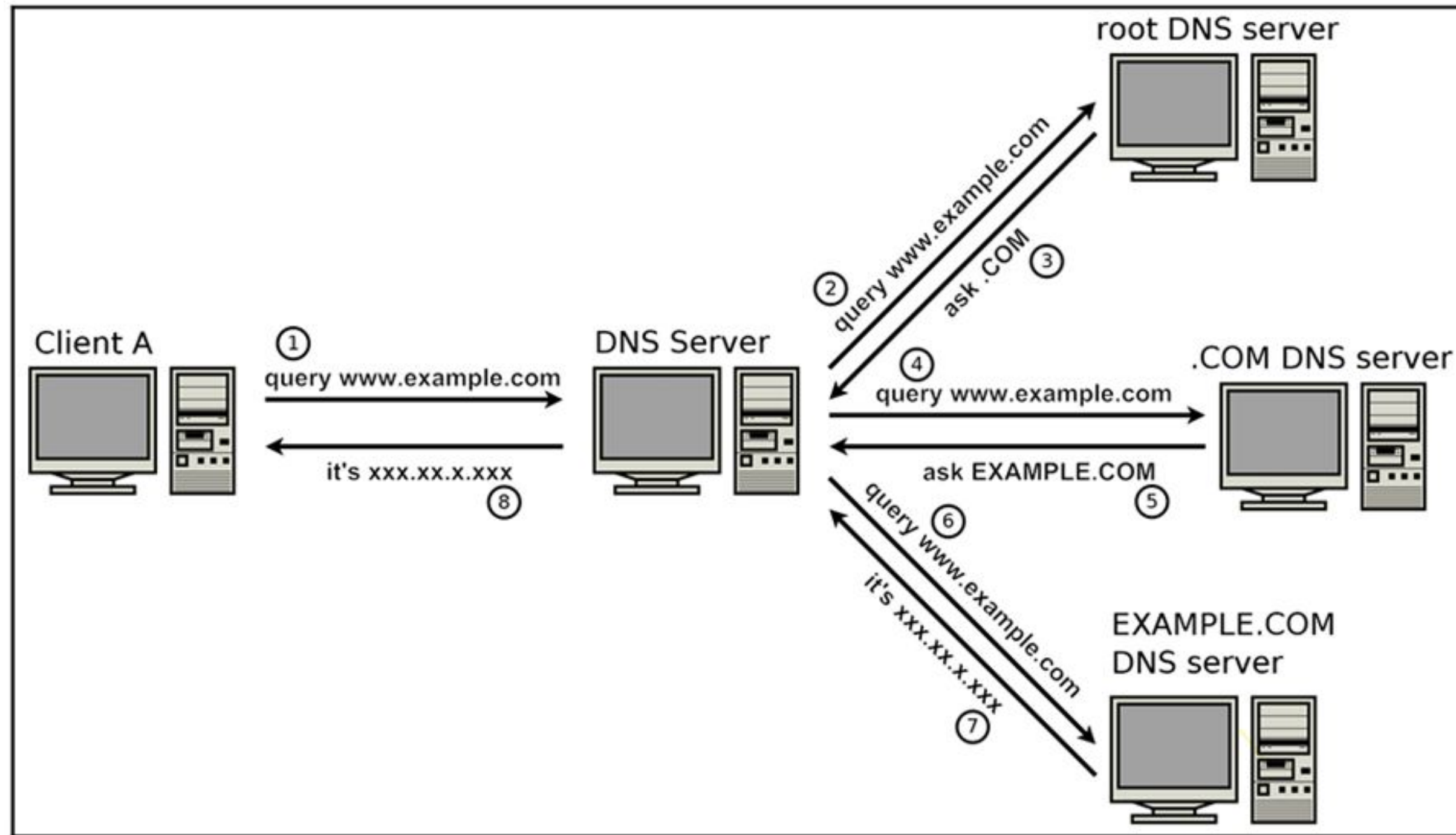
En primer lugar, el SO verifica si ya conocemos la dirección IP, como se observa en la figura siguiente, esto puede pasar si usamos recientemente ese nombre de dominio, el SO admite almacenar esto en la cache local solo durante un tiempo

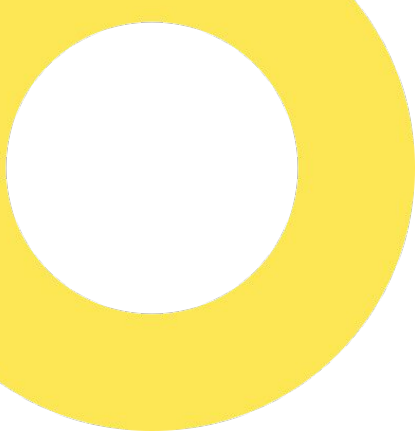


A este tiempo se le llama TTL (Time-to-Live) y es responsabilidad del DNS delimitar ese tiempo. Por otro lado, cuando el nombre del dominio está en la cache local, es obligación del DNS hacer una petición. Comúnmente el servicio DNS es dado por nuestro proveedor de servicio de internet.



# Resolucion de un nombre de dominio





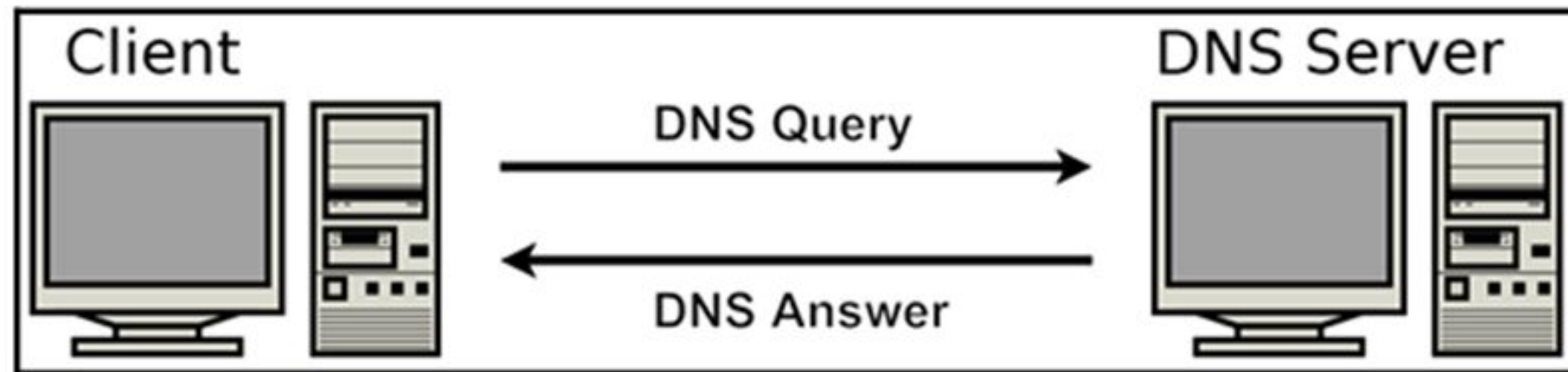
# DNS RECORD TYPES



Record Type	Type ID (decimal)	Description
A	1	IPv4 address record
AAAA	28	IPv6 address record
MX	15	Mail exchange record
TXT	16	Text record
CNAME	5	Canonical name
*	255	All cached records



# DNS PROTOCOL

La forma en que DNS hace consultas es por medio de UDP usando el puerto 53, de esta forma, y si es posible, devuelve una respuesta.





Si la petición es demasiado larga para empaquetarse en un UDP, entonces este paquete se puede enviar a través de TCP con un tamaño de 16 bits, esto se llama TCP fallback o DNS transport over TCP


24

Aunque el protocolo mas ampliamente usado para el envio de consultas es UDP


La cabecera del mensaje tiene una longitud de 12 bytes y es el mismo para la consulta y la respuesta del DNS

DNS Message Format	
Header	Information about the message
Question	Question for the name server
Answer	Answer(s) to the question
Authority	Pointers to other name servers
Additional	Additional information

# DNS MESSAGE FORMAT



El nombre del servidor consiste en dos valores de 16 bits del tipo QTYPE y QCLASS. Estos tipos de datos indican el tipo de registro por el cual se esta preguntando y QCLASS es 1 cuando es desde internet



El formato de la respuesta esta constituido de los mismos campos que las preguntas donde, grosso modo nos indica el tiempo que se puede guardar la respuesta y los tipos de datos enviados

# Conclusiones

LAS APLICACIONES NO ORIENTADAS A CONEXIÓN PERMITEN UNA COMUNICACIÓN EFICIENTE SIN LA NECESIDAD DE ESTABLECER UNA CONEXIÓN PREVIA. ESTO ES ÚTIL EN SITUACIONES DONDE LA VELOCIDAD ES MÁS IMPORTANTE QUE LA PRECISIÓN, COMO EL STREAMING DE VIDEO O AUDIO.

EL PROTOCOLO FTP, POR OTRO LADO, ES VITAL PARA LA TRANSFERENCIA DE ARCHIVOS EN LA RED. PERMITE A LOS USUARIOS CARGAR Y DESCARGAR ARCHIVOS, LO QUE FACILITA EL INTERCAMBIO DE INFORMACIÓN.

FINALMENTE, EL DNS JUEGA UN PAPEL CRUCIAL EN LA NAVEGACIÓN EN LA WEB, TRADUCIENDO LOS NOMBRES DE DOMINIO FÁCILES DE RECORDAR EN DIRECCIONES IP QUE LAS MÁQUINAS PUEDEN ENTENDER.

POR LO TANTO, ESTOS TRES ELEMENTOS JUNTOS PERMITEN UNA COMUNICACIÓN Y NAVEGACIÓN EFICIENTES EN LA RED, FACILITANDO EL INTERCAMBIO DE INFORMACIÓN Y RECURSOS.

# Referencias

- [1] VAN WINKLE, L. (2019). HANDS-ON NETWORK PROGRAMMING WITH C (P. 25, 44-45, 47, 50, 51, 140-146 ).  
LIVERY PLACE 35 LIVERY STREET BIRMINGHAM B3 2PB, UK.: RICHA TRIPATHI. LIVERY PLACE 35 LIVERY STREET  
BIRMINGHAM B3 2PB, UK.: RICHA TRIPATHI.
- [2] HALL, B. (2005). BEEJ'S GUIDE TO NETWORK PROGRAMMING USING INTERNET SOCKETS (P. 4-6, 21 ).
- [3] KATHIRAVELU, P. (2017). PYTHON NETWORK PROGRAMMING COOKBOOK (2.<sup>a</sup> ED., P. 141-148, 179-181, 309,  
311-315, 318-320). PACKT PUBLISHING LTD. LIVERY PLACE 35 LIVERY STREET BIRMINGHAM B3 2PB, UK. PACKT  
PUBLISHING LTD. LIVERY PLACE 35 LIVERY STREET BIRMINGHAM B3 2PB, UK.