**Objetivo:** Investigar todo el conocimiento necesario para lograr la comunicación entre dos servidores con diferentes sistemas operativos, Linux y Windows.

**Prioridad:** Usar un canal seguro, automatizado y con comunicación protegida

**Requisitos:** Protocolos de transferencia de archivos seguros, mecanismos de autenticación robustos y tecnicas de encriptación que garanticen la confidencialidad y la integridad de los datos transmitidos.

**Asegurarse de que**: los procesos de envio se lleven acabo de una manera completamente automatizada sin intervención manual (Herramientas: Scripts, Schedule Tasks o Orchestration Tools).

**Pregunta 1: ¿Que son los canales de transmición en red?**

**Se encontró que**: Un canal de transmisión es el medio lógico o físico por el que viajan lo datos entre dos puntos de una red. Es el “camino” que permite que un emisor y un receptor intercambien información con ciertas reglas, capacidades y niveles de confidencialidad.

**!Vamos a entenderlo por capas¡**

1. **Capa Física**: el soporte por donde se mueven las señales. Cobre(Ethernet), fibra óptica, radio (Wifi).
2. **Capa de Enlace y red**: cómo se encamina y entrega la información. Tramas Ethernet, direcciones IP, rutas, VLAN, MPLS.
3. **Transportes**: el “conducto” extremo a extremo entre procesos. TCP(confiable, orientado a conexión). Un “socket TCP” es un canal lógico entre dos aplicaciones.
4. **Aplicación**: protocolos que usan ese canal. (HTTP(S), SSH, SMB, SFTP, RDP, etc.).

**Pregunta 2: ¿Cuales son las funciones, tipos y diferencias de los canales por capas?**

**Canales Capa Física**

La capa física define cómo se representan, transmiten y reciben los bits sobre un medio. Sus “canales” son los medios y tecnologías que convierten bits en señales eléctricas, ópticas o de radio.

**Funciones principales**

Su principal función es la representación de bits en codificación de línea y niveles de señal, transmitiendolos como señal eléctrica, óptica o radiofrecuencia. Se encargan de sincronizar y temporalizar la transmición para asegurar que el receptor pueda identificar correctamente cuando comienza y termina cada bit. Se maneja un control eléctrico que regula el voltaje, la impedancia y la potencia de transmisión. La conectividad y pinout se basan en conectores, pines, pares y polaridad.

Soportan la detección de errores de bajo nivel ante la pérdida de portadora, pérdida de señal y errores físicos. Implementan soporte de duplexidad: simplex, half-duplex o full-duplex según el medio/tecnología.

**Tipos de canales de la capa física y sus características**

1. **Cobre (par trenzado, RJ-45):** es un cable que contiene holis trenzado por par, UTP/STP. Categorías Cat5e, Cat6A, Cat7/8. Cada una de estas categorías soportan margenes de velocidad y distancias típicas:
   * Cat5e: 1GbE hasta 100 m. 2.5/5 GbE en condiciones favorables.
   * Cat6: 10 GbE hasta ~55 m. 1 GbE hasta 100 m.
   * Cat6A: 10 GbE hasta 100 m.

Pros: económico, fácil de instalar, PoE, muy común.

Contras: susceptible a EMI, limitaciones de distancia para 10 GbE, volumen de cableado.

Usos típicos: LAN de oficina y datacenter a corta distancia.

1. **Cobre (coaxial):** conductor central y malla, mejor blindaje que UTP. Las velocidades y el alcance dependen del estándar; hoy más común en TV/ISP y radiofrecuencia.

Pros: buena inmunidad a ruido.

Contras: menos usado para LAN modernas, conectores y manejo más engorrosos.

1. **Fibra óptica: es una guía de luz. Multimodo (OM3/OM4) vs Monomodo (OS1/0S2).** Las velocidades y el alcance varían según el modo:
   * Multimodo: 10-100 Gb/s a centenares de metros.
   * Monomodo: 1-400 Gb/s a kilómetros o decenas de km.
2. **Radio inalámbrico (Wi-Fi, microondas, 4G/5G, enlaces punto a punto)**
   * Wi-Fi:
     + Bandas 2.4/5/6 GHz, MIMO/OFDM.
     + Pros: movilidad y despliegue rápido.
     + Contras: contención, interferencias, menor estabilidad y latencia variable.
   * Microondas punto a punto:
     + Requiere línea de vista. Enlaces de km con alta disponibilidad si bien diseñados
   * Celular (4G/5G):
     + Cobertura amplia. Útil como respaldo.
     + Contras: jitter/latencia variable, costos de datos.

Los usos típicos son brindar acceso de usuarios, enlaces temporales, redundancia WAN.

* + **Infrarrojo y otros (menos comunes hoy día)**
    - Requieren línea de vista estricta para casos muy específicos.

**Diferencias prácticas entre los principales canales**

La fibra óptica se destaca por tener una mayor capacidad y alcance ante el cobre y Wi-Fi.

Con el riezgo de EMI (Interferencia electromagnética), la fibra no presenta vulnerabilidad.

El Coaxial presenta una mayor resistencia en comparación con el par trenzado y el Wi-Fi el cual es más vulnerable a las interferencia.

Con el margen de latencia y estabilidad, la fibra al igual que el cobre son muy estables, superando las microondas y las redes Wi-Fi/4G/5G siendo estos últimos menos estables.

A nivel de costos, el par trenzado suele ser más barato y fácil de conseguir. La fibra exige herramientas y técnicos.

En la seguridad física la fibra se destaca ya que es difícil de pinchar sin cortar o degradar el enlace. El cobre es más vulnerables. El Wi-Fi es “aire” y depende del cifrado para la privacidad.

En términos de movilidad solo inalámbricos permiten movimiento del endpoint sin recablear.

**Canales Capa Enlace y Red**

**Funciones principales**

**Capa de enlace**: esta capa se encarga de encapsular bits en tramas y controlar el acceso al medio, se encarga del direccionamiento local mediante direcciones MAC. Detecta errores de trama (CRC) y control de flujo entre nodos vecinos.

Proporciona segmentación del dominio de colisión y, con switches, dominio de broadcast por VLAN.

Tecnologías típicas: Ethernet (802.3), Wi-Fi (802.11), PPP, HDLC, VLAN 802.1Q, STP/RSTP, LACP.

**Capa de red**: esta encapsula tramas en paquetes y los enruta entre redes. Provee direccionamiento lógico y jerárquico con IP (IPv4/IPv6).

Selección de ruta, reenvío y control de congestión básico.

Fragmentación/MTU en IPv4, encabezados y extensiones en IPv6.

Protocolos: IP, ICMP/ICMPv6, ARP/ND, OSPF, BGP, RIP, PIM; túneles como IPsec, GRE.

**“Canales” típicos en cada capa**

**Capa enlace**: Ethernet conmutado por puertos y VLANs como canales lógicos aislados.

Enlaces punto a punto (PPP) sobre serial, xDSL o radio.

Enlace inalámbrico 802.11 con SSID y seguridad L2(WPA2/3).

Agregación de enlaces (LACP) como canal lógico de mayor capacidad.

**Capa de red**: Subredes IP separadas por routers.

Túneles L3: IPsec, GRE, WireGuard que crean canales virtuales entre redes.

MPLS/VPNs como “capa 2.5” para transportar L2 o L3 con ingeniería de tráfico.

**Características clave**

Capa de enlace: su segmento es local o dominio de broadcast.

Unidades: tramas. MTU típica 1500 bytes en Ethernet.

Fiabilidad: detección de errores (CRC). Retransmisión no estándar a L2 salvo protocolos específicos.

Aislamiento: VLANs separan tráfico sin cambiar IPs.

Conmutación: decisión basada en MAC y tabla de aprendizaje.

**Capa de red**: su alcance es de extremo a extremo entre redes distintas.

Unidades: paquetes. Soporta MTU discovery y fragmentación (IPv4).

Fiabilidad: IP es no fiable; se apoya en capas superiores o túneles seguros.

Aislamiento y seguridad: subredes, ACLs, listas de rutas, NAT, firewalls.

Enrutamiento: decisión basada en prefijos IP y tabla de rutas.

**Diferencias esenciales L2 vs L3**

* Identidad
  + L2: MAC plana, local al segmento.
  + L3: IP jerárquica, globalmente enrutable.
* Alcance del canal
  + L2: dominio de broadcast o VLAN.
  + L3: entre subredes a través de routers.
* Control del tráfico
  + L2: switching, STP, QoS L2 (802.1p).
  + L3: routing dinámico, políticas, QoS por DSCP.
* Aislamiento
  + L2: VLAN etiqueta tramas.
  + L3: subredes separadas, rutas, ACLs y NAT.
* Seguridad
  + L2: port security, 802.1X, protección contra MAC flood
  + L3: IPsec, listas de acceso, filtrado por IP/puerto, anti-spoofing.
* Túneles
  + L2: QinQ, VXLAN (L2 sobre L3) extienden dominios de broadcast.
  + L3: IPsect/GRE/WireGuard crean canales cifrados entre redes.

**Canales de la capa de Transporte**

En transporte, “canal” se entiende como la conexión lógica extremo a extremo entre procesos, identificada por ***sockets IP:puerto.*** Los principales son TCP, UDP y SCTP. Hoy también se usa QUIC (funciona sobre UDP).

**Funciones de la capa de transporte**

Se encarga de tareas como multiplexación y demultiplexación que procesan múltiples flujos por una misma IP usando puertos.

Segmentación y reensamblaje para dibidir datos de la aplicación en segmentos.

Su fiabilidad y orden asegura la entrega íntegra y en orden cuando aplica.

Control de flujo que adapta la tasa al receptor.

Control de congestión que adapta la tasa a la red.

Establecimiento y cierre de conexión con handshakes y liberación de recursos.

**Canales principales y sus caraterísticas**

* TCP (Transmission Control Protocol)
  + Orientado a conexión. Handshake de 3 vías.
  + Fiable: retransmite, garantiza orden y entrega sin duplicados.
  + Control de flujo: ventanas (rwnd).
  + Control de congestión: algoritmos como Reno, CUBIC, BBR.
  + Stream de bytes: la app define los límites lógicos.
  + MSS/MTU: ajuste para evitar fragmentación.
  + Compatible con NAT y firewalls. Muy extendido.
  + Uso típico: HTTP(S), SSH, SMTP, IMAP/POP, SMB, SQL, etc.
* UDP (User Datagram Protocol)
  + No orientado a conexión. Sin fiabilidad ni orden garantizados.
  + Datagrams: preserva límites de mensaje.
  + Mínima latencia y overhead.
  + No tiene control de flujo/congestión integrado.
  + Ideal para tiempo real y protocolos propios en capa de app.
  + Uso típico: DNS, VoIP, streaming, juegos, QUIC, NTP, syslog, TFTP.
* SCTP (Stream Control Transmission Protocol)
  + Fiable y con orden por “streams” paralelos (multistreaming).
  + Multihoming: múltiples IPs por extremo para resiliencia.
  + Mensajería con preservación de límites (como UDP) pero fiable (como TCP).
  + Menos desplegado que TCP/UDP; soporte variable.
  + Uso típico: señalización telco, algunas apps industriales.
* QUIC (sobre UDP)
  + Seguridad integrada (TLS 1.3 embebido).
  + 0-RTT/1-RTT: establecimiento muy rápido.
  + Multiplexación sin head-of-line blocking a nivel de flujo.
  + Migración de conexión (cambia IP sin cortar la sesión).
  + Uso típico: HTTP/3, servicios modernos web y móviles.

**Diferencias clave entre los canales**

* Conexión
  + TCP/SCTP/QUIC: requieren establecimiento de estado.
  + UDP: sin conexión.
* Fiabilidad y orden
  + TCP/SCTP/QUIC: fiables; TCP orden global, SCTP/QUIC por flujo.
  + UDP: mejor esfuezo, sin orden.
* Límite de mensaje
  + UDP/SCTP/QUIC: preservan mensajes.
  + TCP: stream continuo (la app)
* Rendimiento/latencia
  + UDP/QUIC: menor latencia de control; QUIC evita bloqueo por cabecera.
  + TCP: más overhead por fiabilidad y control de congestión.
* Resiliencia
  + SCTP: multihoming nativo.
  + QUIC: migración de conexión.
  + TCP/UDP: una sola 5-tuple (IP:puerto) por conexión.
* Compatibilidad operativa
  + TCP/UDP: soporte universal.
  + QUIC: muy extendido ya en web, pero puede topar middleboxes antiguos.
  + SCTP: soporte limitado en redes y sistemas.

**Canales de la capa de Aplicación**

En aplicación, el “canal” es el protocolo que define cómo las aplicaciones intercambian datos y significado sobre el transporte. Ejemplos: HTTP/HTTPS, SSH, SFTP, SMB, RDP, DNS, SMTP/IMAP, MQTT, gRPC.

**Funciones de la capa de Aplicación**

Realiza una semántica y da formato a los para definir mensajes, encabezados, códigos de estado y cómo interpretar respuestas.

Negociación de capacidades: versiones, compresión, formatos (p. ej., HTTP/2 vs HTTP/3, ALPN).

Autenticación y autorización: usuarios, claves, certificados, tokens.

Seguridad de aplicación: cifrado extremo a extremo cuando integra TLS o va sobre SSH, prevención de replay, integridad.

Descubrimiento y direccionamiento lógico: nombres, rutas, endpoints, recursos.

Control de sesión: inicio, mantenimiento y cierre de sesiones o canales lógicos.

**Características comparables entre “canales” de aplicación**

* Estado
  + Stateless: cada petición indepentiente (HTTP clásico)
  + Stateful: sesión persistente con contexto (SSH, SMB, RDP).
* Transporte subyacente
  + TCP, UDP, QUIC. Algunos funcionan sobre varios (HTTP/3 sobre QUIC, DNS puede ser UDP/TCP/HTTPS/QUIC).
* Seguridad
  + Integradas: HTTPS/TLS, HTTP/3 (TLS en QUIC), SSH, SFTP.
  + Externa: protocolo sin cifrar protegido por VPN/IPsec.
* Modelao de datos
  + Texto humano-legible (HTTP, SMTP) vs binario eficiente (gRPC, SMB2 +, RDP).
* Patrones de uso
  + Petición-respuesta (HTTP), flujo interactivo (SSH), transferencia de archivos (SFTP, SMB), escritorio remoto (RDP), mensajería pub/sub (MQTT).
* Redimiento
  + Multiplexación y compresión (HTTP/2/3, SMB3), reanudación de sesión, control de ventana a nivel de app.
* Interoperatividad
  + Estándares abiertos y ubicuos (HTTP/HTTPS, DNS, SMTP) vs ecosistemas más cerrados o dependientes de OS (SMB, RDP).
* Compatibilidad con cortafuegos/NAT
  + “Amigables” suelen usar 443/TCP o 443/UDP (HTTPS/HTTP3); otros pueden requerir puertos específicos (445 SMB, 22 SSH, 3389 RDP).

**Diferencias prácticas entre canales comunes**

* HTTP vs HTTPS vs HTTP/3
  + HTTP: texto, stateless, sobre TCP 80, sin cifrado.
  + HTTPS: HTTP sobre TLS en TCP 443. Autenticación por certificados, soporte a OAuth/JWT.
  + HTTP/3: HTTP sobre QUIC (UDP 443), menor latencia, multiplexación sin bloqueo por cabecera.
  + Uso: APIs, web, automatización vía curl, PowerShell Invoke-WebRequest.
* SSH vs SFTP vs SCP
  + SSH: canal seguro y autenticado para comandos y túneles.
  + SFTP: protocolo de archivos sobre SSH, fiable y scriptable.
  + SCP: legado, rápido pero menos flexible.
  + Uso: administración, transferencia automatizada con claves
* SMB 2/3
  + Protocolo de archivos nativo en Windows; soportado en Linux vía Samba.
  + Cifrado y firma en SMB3, multicanal, compresión.
  + Uso: compartición y mapeo de unidades entre Windows y Linux.
* RDP
  + Escritorio remoto sobre TCP/UDP 3389, puede cifrar y autenticar a nivel de protocolo.
  + Uso: administración gráfica en Windows.
* DNS vs DoT vs DoH
  + DNS “clásico” UDP/TCP 53 sin cifrar.
  + DoT (DNS over TLS) y DoH (DNS over HTTPS): privacidad y autenticidad del canal.
* SMTP/IMAP/POP con STARTTLS o SMTPS/IMAPS/POPS
  + Correo. Soportan canales cifrados y autenticados para tránsito seguro.
* MQTT/AMQP.

**Pregunta 3: Técnicas de encriptación**

**Conceptos base**

* Confidencialidad: que solo quien debe leer pueda leer. Se logra cifrando.
* Integridad: que el mensaje no sea alterado. Se logra con autenticación de mensajes o firmas.
* Autenticación: comprobar identidad de las partes.
* Intercambio de claves: acordar claves efímeras de forma segura, idealmente con Perfect Forward Secrecy (PFS).

**Técnicas y primitivas**

* Cifrado simétrico moderno
  + *AES-GCM 128/256*

AES-GCM (Advanced Encryption Standard - Galois/Counter Mode) es una técnica de cifrado simétrico moderna que combina seguridad, eficiencia y autenticación. Es un modo de operación del algoritmo AES que proporciona:

* + - **Confidencialidad:** mediante cifrado de datos.
    - **Integridad y autenticación:** mediante un código de autenticación (tag).

Funciona con claves de 128, 192 o 256 bits, y opera sobre bloques de 128 bits.

* + *ChaCha20-Poly1305*

ChaCha20-Poly1305 es una técnica de cifrado moderno que combina velocidad, seguridad y autenticación, especialmente diseñada para entornos donde AES no es óptimo. Es un esquema de cifrado autenticado con datos asociados (AEAD) que une:

* + - ChaCha20: un cifrador de flujo rápido y seguro.
    - Poly1305: un código de autenticación de mensajes (MAC) eficiente.

Fue diseñado por Daniel J. Bernstein y estandarzado en RFC 8439.

* Autenticación de mensaje
  + HMAC con SHA-256/384 cuando el protocolo no usa AEAD integrado.

HMAC con SHA-256 o SHA-384 es una técnica de autenticación criptográfica que garantiza la integridad y autenticidad de los datos mediante funciones hash seguras. HMAC (Hash-based Message Authentication Code) es un mecanismo que combina:

* + - Una función hash criptográfica (como SHA-256 o SHA-384).
    - Una clave secreta compartida.

Su propósito es verificar que un mensaje no ha sido alterado y que proviene de una fuente legítima.

* Intercambio de claves con PFS
  + ECDHE sobre curvas seguras (X25519, secp256r1)

ECHE (Elliptic Curve Diffie-Hellman Ephemeral) sobre curvas seguras como X25519 Y secp256r1 es fundamental en criptografía moderna para establecer claves compartidas de forma segura y eficiente. ECDHE es una variante del protocolo Diffie-Hellman que usa curvas elípticas y claves efímeras para:

* + - **Establecer claves compartidas** entre dos partes sin transmitirlas directamente.
    - Garantizar secreto perfecto hacia adelante (forward secrety): si se compromete una clave privada, sesiones anteriores siguen seguras.
* Hashes criptográficos
  + SHA-256/384 para integridad y derivación de claves

SHA-256 y SHA-384 son funciones hash criptográficas que juegan un papel crucial en dos áreas clave de la seguridad digital: verificación de integridad y derivación de claves. Las funciones SHA (Secure Hash Algorithm) generan un resumen único de un mensaje. Si el mensaje cambia, el hash cambia drásticamente.

* Firmas digitales
  + Ed25519 o ECDSA P-256 para identidad de servidor/cliente o artefactos.

Ed25519 y ECDSA P-256 son dos algoritmos de firma digital basados en curvas elípticas, ampliamente usados para autenticar identidades (servidores, clientes, artefactos, firmware, etc.). Ambos ofrecen seguridad robusta, pero tienen diferencias importantes en diseño, rendimiento y filosofía criptográfica. Ambos algoritmos permiten:

* + - Verificar la identidad de quien firma un mensaje.
    - Garantizar la integridad del mensaje firmado.
    - Evitar suplantación: solo quien posee la clave privada puede firmar.

**Protocolos que ya integran lo anterior**

* TLS 1.2+ Y TLS 1.3
  + Suites fuertes: TLS\_AES\_128\_GCM\_SHA256, TLS\_AES\_256\_GCM\_SHA384, TLS\_CHACHA20\_POLY1305\_SHA256.
  + PFS por defecto en TLS 1.3. Usar certificados con SHA-256 y claves ECDSA o RSA-2048+.
* SSH 2.0
  + Claves de host y usuario Ed25519.
  + Cifrados: [chacha20‑poly1305@openssh.com](mailto:chacha20%E2%80%91poly1305@openssh.com) o [aes128‑gcm@openssh.com](mailto:aes128%E2%80%91gcm@openssh.com).
  + MACs modernos si no se usa AEAD: hmac-sha2-256.
* IPsec/WireGrard
  + IPsec con AES-GCM y IKEv2 con ECDHE.
  + WireGuard usa NoiseIK con Curve25519, ChaCha20-Poly1305, BLAKE2s. Muy simple y rápido.
* SMB 3.x
  + Firma obligatoria y cifrado en tránsito con AES-GCM.
* HTTPS/HTTP3
  + HTTP/2 sobre TLS 1.2+ o HTTP/3 sobre QUIC con TLS 1.3 integrado.