

**Nombre y Apellido : Benavidez Abigail**

**1. ¿Qué es una red de computadoras y cuál es su propósito principal?**

- Una red de computadoras es un conjunto de dispositivos interconectados que pueden comunicarse y compartir recursos, como datos, archivos, impresoras o conexión a Internet. El objetivo principal de una red es permitir la comunicación y la transferencia de información de manera eficiente entre dispositivos.

**2. Diferencias entre una red local (LAN) y una red de área amplia (WAN).**

**LAN (Local Area Network):** Es una red que cubre una área geográfica limitada, como una casa, oficina o edificio. Es rápida y permite la interconexión de dispositivos dentro de un área pequeña.

**WAN (Wide Area Network):** Es una red que abarca una amplia área geográfica, como una ciudad, país o incluso a nivel global. Es más lenta comparada con una LAN y conecta múltiples redes LAN entre sí a través de grandes distancias.

**3. Función de una tarjeta de red (NIC) en una computadora.**

Una tarjeta de red (NIC, Network Interface Card) es un hardware que permite a una computadora conectarse a una red. La NIC convierte los datos de la computadora en señales que pueden ser transmitidas a través de la red y viceversa, permitiendo la comunicación con otros dispositivos.

**4. Descripción del modelo OSI y sus siete capas. Propósito de cada capa:**

**Capa 1: Física:** Transmite bits a través de medios físicos como cables o ondas.

**Capa 2: Enlace de Datos:** Gestiona la transferencia de datos entre dos dispositivos conectados directamente, y controla los errores de transmisión.

**Capa 3: Red:** Determina la ruta que los datos deben seguir para llegar a su destino, utilizando direcciones IP.

**Capa 4: Transporte:** Garantiza la transferencia confiable de datos, gestionando errores y controlando el flujo.

**Capa 5: Sesión:** Administra las conexiones (sesiones) entre dispositivos.

**Capa 6: Presentación:** Traduce, cifra y comprime los datos.

**Capa 7: Aplicación:** Proporciona servicios de red a las aplicaciones de usuario, como correo electrónico o transferencia de archivos.

#### **5. Comunicación entre dos dispositivos utilizando el modelo OSI:**

Cuando un dispositivo envía datos a otro, los datos se encapsulan a medida que descienden por las capas del modelo OSI, comenzando desde la capa de aplicación hasta la capa física. Los datos viajan a través del medio físico hacia el dispositivo receptor, donde el proceso de desencapsulación ocurre en orden inverso desde la capa física hasta la capa de aplicación.

#### **6. Descripción del modelo TCP/IP y sus capas. Diferencias con el modelo OSI:**

**Modelo TCP/IP:**

**Capa 1: Enlace de Datos:** Maneja la transmisión de datos a nivel de hardware.

**Capa 2: Internet:** Equivalente a la capa de red del modelo OSI, gestiona la ruta y el direccionamiento de los datos.

**Capa 3: Transporte:** Similar a la capa de transporte del modelo OSI, garantiza la transferencia de datos de extremo a extremo.

**Capa 4: Aplicación:** Combina las capas de aplicación, presentación y sesión del modelo OSI, proporcionando servicios a las aplicaciones de usuario.

**-Diferencias:** El modelo TCP/IP tiene 4 capas en lugar de 7, y es más práctico, mientras que el modelo OSI es más teórico y detallado.

#### **7. Función de la capa de aplicación en el modelo TCP/IP. Protocolos que trabajan en esta capa:**

La capa de aplicación en el modelo TCP/IP interactúa directamente con las aplicaciones de software que utilizan los usuarios. Proporciona servicios de red, como correo electrónico, transferencia de archivos y navegación web. Protocolos importantes en esta capa incluyen HTTP, FTP, SMTP y DNS.

#### **8. Descripción de la capa de transporte en el modelo TCP/IP. Principales protocolos:**

La capa de transporte garantiza la transferencia de datos de extremo a extremo entre dispositivos. Los dos principales protocolos en esta capa son:

**TCP (Transmission Control Protocol):**\*\* Protocolo orientado a la conexión que garantiza la entrega fiable y en orden de los datos.

**UDP (User Datagram Protocol):**\*\* Protocolo no orientado a la conexión, que permite una transmisión más rápida pero sin garantía de entrega.

### **9. Función de la capa de red en el modelo TCP/IP. Protocolo principal:**

La capa de red gestiona la dirección y enrutamiento de los paquetes de datos entre redes diferentes. El protocolo principal en esta capa es el IP (Internet Protocol)

### **10. Función de la capa de enlace de datos en el modelo TCP/IP:**

La capa de enlace de datos gestiona la transmisión de datos entre dos dispositivos directamente conectados. Esta capa se encarga de la detección y corrección de errores y del control de acceso al medio.

### **11. ¿Qué es una dirección IP y por qué es importante en la comunicación de redes?**

Una dirección IP es un identificador único asignado a cada dispositivo en una red que permite identificar y comunicar datos entre dispositivos en Internet o en una red local. Es crucial para enrutar paquetes de datos al destino correcto.

### **12. Diferencia entre una dirección IP estática y una dinámica:**

**IP estática:** Es una dirección IP fija asignada permanentemente a un dispositivo. No cambia con el tiempo.

**IP dinámica:** Es una dirección IP asignada temporalmente por un servidor DHCP y puede cambiar cada vez que el dispositivo se conecta a la red.

### **13. ¿Qué es una máscara de subred y cómo se utiliza en una red?**

Una máscara de subred es un conjunto de bits que separa la parte de la dirección IP que identifica la red de la parte que identifica los dispositivos dentro de esa red. Se utiliza para dividir una red grande en subredes más pequeñas y gestionar eficientemente el direccionamiento IP.

#### **14. Proceso de encapsulación de datos desde la capa de aplicación hasta la capa física en el modelo OSI:**

En el modelo OSI, los datos de la aplicación se encapsulan a medida que pasan por cada capa del modelo. En la capa de aplicación, los datos son organizados y preparados para ser transmitidos. En la capa de transporte, se agregan cabeceras para garantizar la entrega. En la capa de red, se añade la información de dirección IP, y en la capa de enlace de datos, se incluyen las direcciones MAC. Finalmente, en la capa física, los datos son convertidos en señales eléctricas o ópticas para ser transmitidos a través del medio físico.

#### **15. ¿Qué es la concurrencia en programación y por qué es importante en el contexto de redes?**

La concurrencia en programación es la capacidad de un sistema para ejecutar múltiples tareas o procesos al mismo tiempo. En el contexto de redes, es importante porque permite que múltiples conexiones y operaciones de red se manejen simultáneamente, mejorando el rendimiento y la capacidad de respuesta.

#### **16. ¿Qué es un thread en programación y cómo se utiliza para manejar la concurrencia?**

Un thread es la unidad más pequeña de un proceso que puede ser ejecutada independientemente. En programación, los threads se utilizan para manejar la concurrencia al permitir que múltiples partes de un programa se ejecuten de manera simultánea, aprovechando mejor los recursos del sistema.

#### **17. ¿Qué es la sincronización de threads y por qué es importante?**

La sincronización de threads es un mecanismo utilizado para controlar el acceso de múltiples threads a recursos compartidos para evitar condiciones de carrera y asegurar la coherencia de los datos. Es importante porque asegura que los threads no interfieran entre sí y que los datos no se corrompan.

#### **18. Diferencias entre procesos y threads:**

Proceso: Es una instancia de un programa en ejecución, con su propio espacio de memoria. Los procesos son independientes entre sí.

**Thread:** Es una unidad de ejecución dentro de un proceso. Los threads de un mismo proceso comparten el mismo espacio de memoria, lo que facilita la comunicación entre ellos, pero también requiere sincronización para evitar conflictos.

#### **19. Situación en la que utilizarías threads para mejorar el rendimiento de una aplicación en red:**

En un servidor web que maneja múltiples solicitudes de clientes simultáneamente, se podrían utilizar threads para procesar cada solicitud de manera independiente. Esto permitiría que el servidor responda rápidamente a cada solicitud sin que las demás se vean afectadas, mejorando el rendimiento general.

#### **20. Protocolo de red moderno:**

**HTTP/2:** Es una versión mejorada del protocolo HTTP que mejora la velocidad y eficiencia en la transferencia de datos en la web. Introduce características como la multiplexación, que permite múltiples solicitudes y respuestas simultáneas en una única conexión, reduciendo la latencia y mejorando la experiencia del usuario en aplicaciones web modernas.