

15/05/2024

UNIVERSIDAD LAICA Eloy ALFARO DE MANABÍ

NOMBRE: DOMINGUEZ ZAMBRANO MERLY MISTHELLE CURSO: 8^{vo}

ING: SINCIGUANO CHIRIBOGA CESAR AUGUSTO PARALELO: "A"

RESPONDA LAS SIGUIENTES PREGUNTAS:

1) ¿Cuál es la diferencia entre Middleware y Sistemas Operativos?

Son dos conceptos distintos dentro de la informática. El middleware es un tipo de software que actúa como intermediario entre diferentes aplicaciones, permitiendo la comunicación y la interacción entre ellas para que puedan trabajar de manera conjunta de una forma más eficiente. El middleware se encarga de gestionar la comunicación entre las distintas aplicaciones, proporcionando servicios y funcionalidades que facilitan el intercambio de información y datos. Un sistema operativo es un software que actúa como intermediario entre el hardware de un ordenador y las aplicaciones que se ejecutan en él. El sistema operativo se encarga de gestionar los recursos del sistema, coordinar las operaciones de entrada y salida, y proporcionar una interfaz de usuario para que los usuarios puedan interactuar con el ordenador de manera cómoda y sencilla.

La diferencia principal entre middleware y sistemas operativos es que el middleware actúa como intermediario entre aplicaciones, mientras que los sistemas operativos actúan como intermediarios entre el hardware y las aplicaciones.

2) Arquitectura de los Sistemas Distribuidos.

Pueden ser utilizadas para diseñar y desarrollar aplicaciones distribuidas.

1. Cliente-servidor: Es la arquitectura más común en la que un servidor central proporciona recursos y servicios a múltiples clientes que se conectan a través de una red.
2. Peer-to-peer (P2P): Todos los nodos o dispositivos de la red tienen capacidades iguales y pueden actuar tanto como clientes como servidores, compartiendo recursos y servicios entre ellos de manera descentralizada.
3. Arquitectura de capas: Se divide la aplicación en varias capas o niveles, donde cada capa es responsable de realizar funciones específicas. Las capas pueden comunicarse entre sí a través de protocolos definidos.
4. Arquitectura de microservicios: La aplicación se descompone en módulos independientes y autónomos que se comunican entre sí a través de APIs. Cada microservicio es responsable de realizar una función específica y puede ser desarrollado y desplegado.
5. Arquitectura basada en eventos: Los componentes del sistema se comunican entre sí a través de eventos, en lugar de llamadas directas. Esto permite

