**Se basan en los pdf del duoc del 1.1.2 a 1.1.3**

**1. Definición y Componentes Clave**

* **Sistema Operativo (SO):**  
  Software intermediario entre hardware y usuario. Gestiona recursos (CPU, memoria, periféricos) y garantiza integridad del sistema.
* **Componentes Principales:**
  + **Kernel:**
    - Gestión de procesos, memoria, dispositivos, sistema de archivos, E/S.
    - Funciona en **modo privilegiado** (acceso total al hardware).
  + **Interfaz de Usuario (UI):**
    - GUI (gráfica) y CLI (línea de comandos).
  + **Gestión de Seguridad:** Autenticación, autorización, auditoría.

**2. Sistemas Operativos Relevantes**

| **Familia** | **Ejemplos** | **Características** |
| --- | --- | --- |
| **Código Cerrado** | Windows, macOS | Optimizados para hardware específico. |
| **Código Abierto** | Linux (Debian, Red Hat) | Personalizable, usado en servidores y IoT. |
| **Móviles** | Android (basado en Linux) | Dominante en smartphones. |

**3. Kernel: Funciones Detalladas**

* **Gestión de Procesos:**
  + Asignación de tiempo de CPU, prioridades y comunicación entre procesos.
  + **Multitarea:** Ejecución simultánea de procesos (ej: navegador + editor de texto).
  + **Multiprocesamiento:** Uso de múltiples núcleos de CPU.
* **Memoria:**
  + **RAM:** Almacenamiento temporal rápido.
  + **Memoria Virtual:** Extensión de RAM usando disco duro.
  + **Cache:** Acelera acceso a datos frecuentes.
* **Sistema de Archivos:**
  + **NTFS (Windows):** Permisos, cifrado.
  + **APFS (macOS):** Optimizado para SSD, snapshots.
  + **ext4 (Linux):** Journaling, alta capacidad.
  + **exFAT:** Universal para almacenamiento flash.

**4. Seguridad y Gestión de Usuarios**

* **Autenticación:** Verificación mediante credenciales (usuario/contraseña).
* **Autorización:** Control de acceso a recursos (ej: archivos, directorios).
* **Roles y Permisos:** Asignación jerárquica (ej: administrador vs. usuario estándar).
* **Auditoría:** Registros de eventos para detectar actividades sospechosas.

**5. Virtualización**

* **Definición:** Creación de entornos virtuales sobre hardware físico.
* **Hipervisor:** Software que gestiona máquinas virtuales (VM).
  + **Ejemplo:** Un servidor físico dividido en VMs para correo, web y apps heredadas.
* **Ventajas:**
  + Optimización de recursos (ej: usar 90% de capacidad vs. 30%).
  + Aislamiento de fallos y flexibilidad en despliegues.

**6. Actividades Prácticas (Estilo MIT/Harvard)**

1. **Análisis Comparativo:**
   * En grupos, investiguen Windows, Linux y macOS.
   * Compare: arquitectura del kernel, sistemas de archivos, gestión de memoria.
2. **Caso de Estudio:**
   * Diseñen un escenario donde la virtualización resuelva un problema de infraestructura (ej: consolidación de servidores).
3. **Debate:**
   * Discuta: ¿Cómo afecta la elección del SO a la seguridad empresarial?

**7. Preguntas Clave para Reflexión**

* ¿Por qué el kernel es considerado el "corazón" del SO?
* ¿Cómo impacta la memoria virtual en el rendimiento de aplicaciones pesadas?
* ¿Qué ventajas tiene Linux en entornos de nube vs. Windows?

**Recursos Adicionales:**

* Libro: *Operating System Concepts* (Silberschatz, Galvin, Gagne).
* Herramientas: VirtualBox (para prácticas de virtualización), Wireshark (análisis de red).

**© Elaborado por: [Tu Nombre] | Nivel: Expertos en TI**  
*Nota: Esta guía integra estándares pedagógicos de instituciones líderes para un aprendizaje profundo y aplicado.*