TRABAJO PRÁCTICO N.º 5
SISTEMAS OPERATIVOS, ARQUITECTURA DE SOFTWARE Y COMPUTACIÓN EN LA NUBE

Sussini Patricio

EJERCICIO 1: MÉTODOS DE INSTALACIÓN DE SOFTWARE EN LINUX

Completa la tabla seleccionando el método de instalación adecuado según la descripción dada.

Descripción	Método de Instalación
Permite instalar paquetes desde repositorios oficiales con resolución automática de dependencias.	Gestores de paquetes
Ejecuta archivos binarios precompilados sin necesidad de instalación.	Ejecutables binarios
Facilita la instalación de paquetes autocontenidos.	Snap, Flatpak y AppImage
Se basa en la descarga del código fuente y su compilación manual.	Código fuente

EJERCICIO 2: MODELOS DE ARQUITECTURA DE SOFTWARE

- Integra todas las funciones en un solo bloque de código. → Monolítica
- Se basa en la comunicación entre un cliente y un servidor centralizado. → Cliente-Servidor
- Divide la aplicación en servicios pequeños e independientes. → Microservicios
- Ofrece facilidad de desarrollo y despliegue para proyectos pequeños. → Monolítica

EJERCICIO 3: COMPUTACIÓN EN LA NUBE Y MODELOS DE SERVICIO

- Un usuario accede a Google Docs desde su navegador sin instalar software adicional. → SaaS
- Una empresa aprovisiona máquinas virtuales en AWS EC2 y las configura manualmente. → IaaS

- Un desarrollador sube su código a Google App Engine, que asigna recursos automáticamente. → PaaS
- Un banco gestiona sus propios servidores privados para cumplir con normativas de seguridad. → Ninguno (infraestructura on-premise)

EJERCICIO 4: ESTRATEGIAS DE DESPLIEGUE EN LA NUBE

- Nube Pública: Infraestructura compartida gestionada por proveedores como AWS o Google Cloud.
- Nube Privada: Recursos dedicados a una única organización con mayor control y seguridad.
- Nube Híbrida: Combina recursos privados y públicos para optimizar costos y flexibilidad.
- Multinube: Uso de múltiples proveedores de nube para evitar dependencia de un solo servicio.

EJERCICIO 5: VENTAJAS Y DESAFÍOS DE LOS MODELOS DE ARQUITECTURA

- Facilita la escalabilidad independiente de cada servicio. → Ventaja
- Aumenta la complejidad en la comunicación entre servicios. → Desafío
- Permite el despliegue individual de cada componente. → Ventaja
- Requiere herramientas adicionales para la gestión y monitoreo. → Desafío