



Fundamentos – Componentes de los Sistemas de TI

Ciberseguridad – FPUNA



Arquitectura de Sistemas

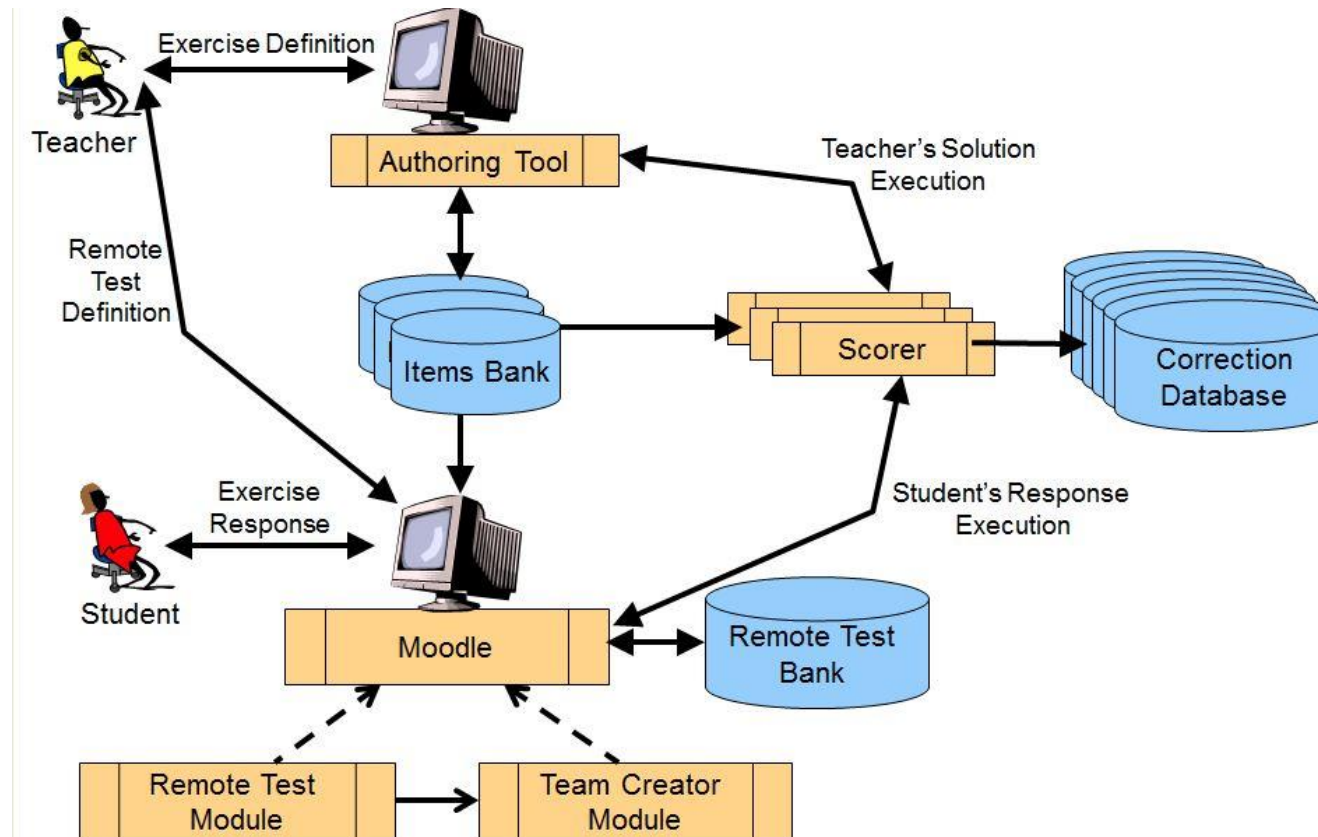
Arquitectura de Sistema

- Una arquitectura de sistema es el modelo conceptual que define la estructura, el comportamiento y más vistas de un sistema. Una arquitectura es una descripción y representación formal de un sistema, organizada de forma que se soporte sobre las estructuras y comportamientos del sistema.
- La arquitectura de un sistema puede consistir en componentes del sistema y los subsistemas desarrollados, que trabajarán juntos para implementar el sistema global.

Algunos aspectos de diseño de Arquitectura de Sistema

- Los atributos de calidad del sistema.
- El entorno de TI de la empresa.
- El diseño del sistema es lo que quiere el cliente.
- El sistema debe cumplir con las estrategias comerciales.
- Dinámica humana, significa que si el sistema se entrega a personas sin conocimientos técnicos, entonces el sistema debería ser auto-mantenible hasta cierto punto.

Ejemplo de una Arquitectura de Sistema



Arquitectura de Computadora

Es la organización lógica del hardware de las computadoras.

Se trata de un conjunto de principios que describen cómo se pueden describir las características del hardware de la computadora y cómo interactúan entre sí.

La arquitectura de un computadora determina su rendimiento, sus capacidades y sus límites.

Componentes principales de la arquitectura de computadoras

- **CPU:** es la unidad central de procesamiento o microprocesador. Esta parte se encarga de ir ejecutando las diferentes instrucciones de la ISA y los datos que el software emplea para su ejecución. Es decir, es la encargada de ejecutar los programas informáticos, incluido el sistema operativo.
- **Bus:** se refiere a los componentes que entrelazan partes de la computadora y pueden ser de varios tipos y características, como el bus de datos, el bus de direcciones, y el bus de control.
- **Memoria principal:** es la memoria RAM, generalmente, donde se guardan los programas que se van a ejecutar, es decir, los datos e instrucciones necesarios para un proceso y que serán reclamados por la CPU.
- **E/S:** por supuesto, las computadoras también necesitan un sistema de entrada y salida de la información, es decir, puertos por donde enviar y recibir datos. Esto es fundamental para el usuario, ya que de lo contrario no podría interactuar con la computadora.

Tipos de arquitectura de computadoras

- **Arquitectura de flujo de datos (dataflow):** Es una arquitectura paralela en la que los datos pasan por las diferentes etapas del cálculo. En esta arquitectura, el resultado de un cálculo se utiliza como entrada para otro cálculo.
- **Arquitectura von Neumann:** Es una arquitectura secuencial en la que el cálculo se ejecuta secuencialmente. El cálculo se realiza en un orden secuencial. Esta arquitectura tiene una unidad central de procesamiento (CPU), una memoria y dispositivos de entrada-salida.
- **Arquitectura Harvard:** Es una versión modificada de la arquitectura von Neumann. Tiene dos memorias separadas: una para las instrucciones del programa y otra para los datos.
- **Arquitectura Harvard modificada:** Es una combinación de la arquitectura de flujo de datos y la arquitectura Harvard. En esta arquitectura, hay una memoria de programa y una memoria de datos.

Arquitectura de Sistema Operativo

Para poder construir un Sistema Operativo se deben tener en cuenta dos tipos de requisitos, los cuales son:

- **Requisitos de usuario:** Un sistema fácil de usar y de aprender, seguro, rápido y adecuado para el uso que se le necesita dar.
- **Requisitos del software:** Considera el continuo mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia frente a los errores y flexibilidad.

El objetivo de la estructuración es buscar una organización interna que facilite la comprensión, incremente la portabilidad, extensión y favorecer el mantenimiento de los Sistemas Operativos.



Arquitecturas de Sistemas Operativos

- Arquitectura monolítica.
- La arquitectura de Micronúcleo o Microkernel.
- La arquitectura por capas o jerárquica (Estructura por niveles).
- Arquitecturas por módulos.

Arquitectura Monolítica (1/2)

Proporciona una máxima funcionalidad dentro del menor espacio posible.

Se caracteriza porque no tienen una estructura totalmente clara con respecto a rutinas y funcionalidades (ej. manejo de drivers, sistemas de archivos, gestión de memoria, etc.), se encuentran agrupados en un solo programa (el Sistema Operativo).

Está descrito como un conjunto de procedimientos o rutinas entrelazadas de tal forma que cada una tiene la posibilidad de llamar a las otras rutinas cada vez que así lo requiera. Sin embargo, cabe destacar las falencias en este tipo de estructura que radica principalmente en la poca confiabilidad otorgada, ya que todo el sistema, al no tener una estructura definida, se ejecuta todo en el mismo nivel del núcleo (kernel) lo que lo hace altamente vulnerable, por esta razón cuando falla un programa se produce un error en todo el sistema.

Arquitectura Monolítica (2/2)

Además si se modifica el hardware por lo general es necesario recompilar el kernel para poder disponer de las funcionalidades. Esto consume tiempo y recursos porque la compilación de un nuevo kernel puede durar varias horas y necesita de una gran cantidad de memoria.

Cada vez que alguien añade una nueva característica o corrige un error, significa que se necesitará hacer una recompilación del kernel entero, un ejemplo de esto podemos verlo en Linux.

También el hecho de que en el espacio del kernel están incluidos todos los servicios básicos, tiene tres grandes inconvenientes: el tamaño del núcleo, la falta de extensibilidad y la mala capacidad de mantenimiento.



Ejemplos de Sistemas Operativos con Arquitectura Monolítica

- Linux
- Windows

Arquitectura Micronúcleo o Microkernel

Las funciones centrales son manejadas por el núcleo (kernel) y la interfaz de usuario es manejada por el entorno (shell).

El Microkernel se encarga de todo el código de un sistema, y de planificar los hilos (threads) con la finalidad de tener multitareas.

Ventajas de la Arquitectura Micronúcleo

- **Uniformidad de interfaces:** disponen de una interfaz única para las solicitudes de los procesos, el paso de mensajes.
- **Portabilidad:** reduciendo el núcleo e implementando casi todo en servidores, para implementarlo en arquitecturas diferentes, sólo habría que modificar el núcleo haciendo más simple su portabilidad.
- **Fiabilidad:** es más fácil corregir fallas en un sistema pequeño ya que se pueden realizar pruebas más rigurosas que en un sistema mucho más grande.

Ejemplos de Sistemas Operativos con Arquitectura Micronúcleo

- AIX
- BeOS
- Mach
- QNX
- Minix
- Hurd

Arquitectura por capas o jerárquica (Estructura por niveles)

- El Sistema Operativo queda definido modularmente por divisiones en capas o niveles, cuya organización está dada como una jerarquía de capas donde cada una de ellas ofrece una interfaz clara y bien definida, la capa superior solamente utiliza los servicios y funciones que ofrece la capa inferior, es decir, la **capa n** sólo se comunica para obtener lo requerido con la **capa n-1**, donde la capa inferior es la más privilegiada. El encargado de que solamente haya comunicación entre capas adyacentes es el procesador.

Arquitectura por capas o jerárquica (Estructura por niveles)

La arquitectura original consta de 6 capas:

- **Capa 5:** Se encuentra la interfaz de usuario.
- **Capa 4:** Aloja los programas de usuario.
- **Capa 3:** Se controlan los dispositivos E/S (entrada y salida).
- **Capa 2:** Se administra la comunicación inter-proceso y la consola del operador.
- **Capa 1:** Administración de memoria y discos.
- **Capa 0:** Correspondiente al Hardware, realizando asignación del procesador, también alterna entre procesos cuando ocurren interrupciones o se han expirado y proporciona multiprogramación básica de la CPU.

Ejemplos de Sistemas Operativos con Arquitectura por capas

- THE (Technische Hogeschool Eindhoven)
- Venus
- MULTICS (Multiplexed Information and Computing Service)

Arquitectura por módulos

La mayoría de los sistemas operativos modernos implementan este enfoque. Lo que caracteriza este tipo de estructura es que el kernel se compone por módulos, y cada uno de estos módulos se encuentra separado de forma independiente, tal que, si alguno falla no afecta a los otros, ni al núcleo, por ejemplo, si el módulo de software que se encarga de controlar el proceso de Telnet en una unidad se bloquea o es atacado, sólo este proceso se verá afectado. El resto de las operaciones siguen sus funciones habituales.

Los módulos se pueden cargar dinámicamente en el núcleo cuando se necesiten, ya sea, en tiempo de ejecución o durante el arranque del sistema. El kernel dispone de los componentes fundamentales y se conectan directamente con servicios adicionales. Además otros componentes pueden cargarse dinámicamente al núcleo. Este enfoque modular utiliza la programación orientada a objetos.

Arquitectura por módulos

En general, esta estructura se parece bastante a la de capas, pero es mucho más flexible debido a que cualquier módulo de esta estructura puede llamar a otro. Es similar a la estructura de microkernel, pues el kernel también tiene las funciones esenciales, pero este es más eficiente ya que, no necesitan un mecanismo de paso de mensajes para comunicarse, sólo interfaces conocidas.



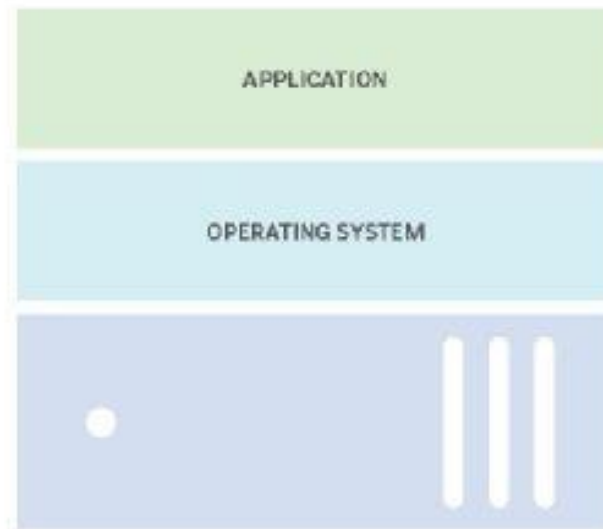
Ejemplos de Sistemas Operativos con Arquitectura por módulos

- Unix modernos
- Solaris
- Linux
- Mac OSX

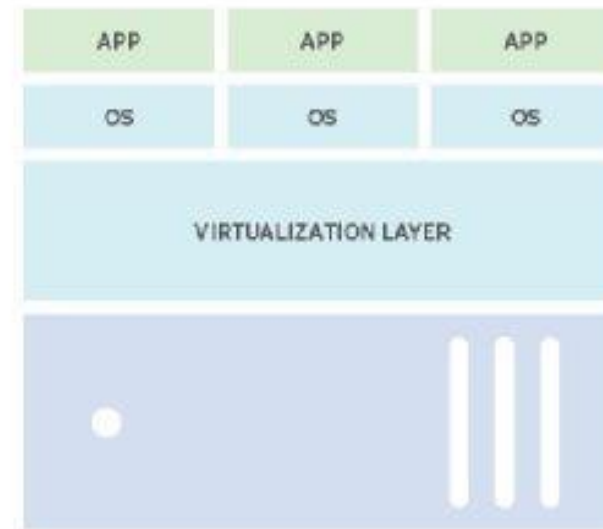
Arquitectura de Virtualización

Una arquitectura de virtualización es un modelo conceptual que especifica la disposición y las interrelaciones de los componentes particulares involucrados en la entrega de una versión virtual, en lugar de física, de algo, como un sistema operativo (SO), un servidor, un dispositivo de almacenamiento o recursos de red.

Traditional and virtual architecture



Traditional architecture



Virtual architecture

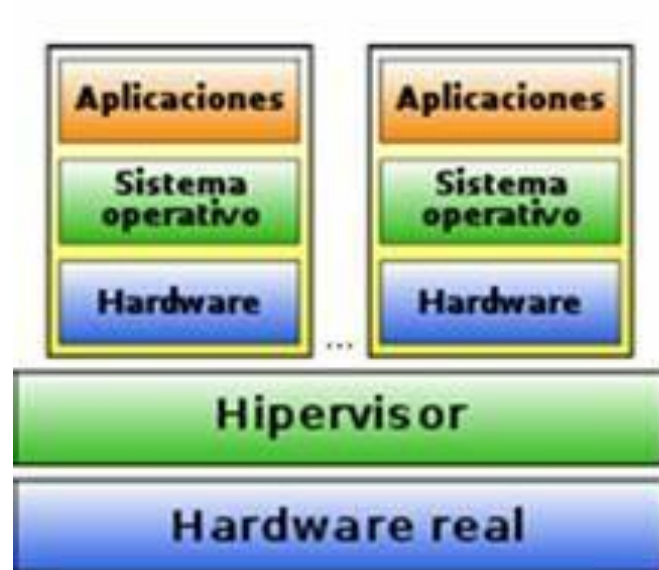
Virtualización e hipervisores

- La virtualización suele estar basada en hipervisores. El hipervisor aísla los sistemas operativos y las aplicaciones del hardware informático subyacente para que la máquina host pueda ejecutar varias máquinas virtuales (VM) como invitados que comparten los recursos informáticos físicos del sistema, como los ciclos del procesador, el espacio de memoria, el ancho de banda de la red, etc.

Tipos de Hipervisores

- **Tipo 1:** a veces denominados hipervisores bare-metal, se ejecutan directamente sobre el hardware del sistema host. Los hipervisores bare-metal ofrecen alta disponibilidad y gestión de recursos. Su acceso directo al hardware del sistema permite un mejor rendimiento, escalabilidad y estabilidad. Ejemplos de hipervisores de tipo 1 incluyen **Microsoft Hyper-V, Citrix XenServer y VMware ESXi.**
- **Tipo 2:** también conocido como hipervisor alojado, se instala en la parte superior del sistema operativo del host, en lugar de sentarse directamente sobre el hardware como lo hace el hipervisor de tipo 1. Cada sistema operativo o VM invitado se ejecuta por encima del hipervisor. La conveniencia de un sistema operativo host conocido puede facilitar la configuración del sistema y las tareas de administración. Sin embargo, la adición de una capa de sistema operativo anfitrión puede potencialmente limitar el rendimiento y exponer posibles fallas de seguridad del sistema operativo. Ejemplos de hipervisores de tipo 2 incluyen **VMware Workstation, Virtual PC y Oracle VM VirtualBox.**

Tipos de Hipervisor



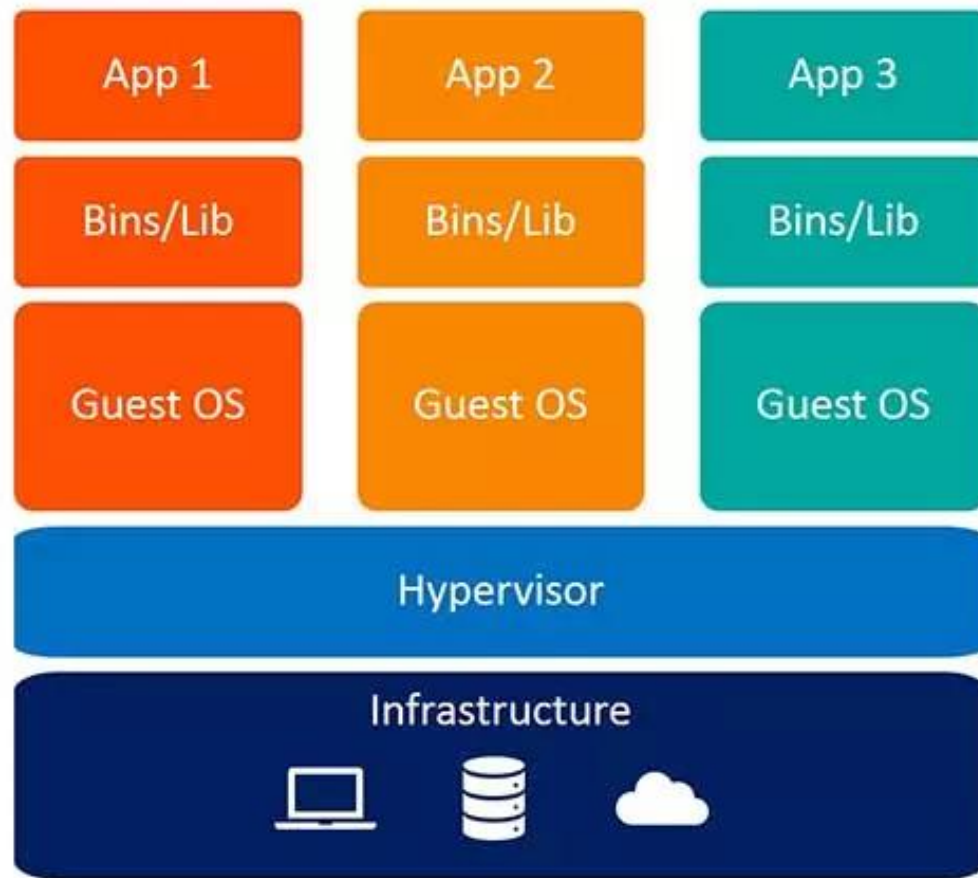
Tipo 1



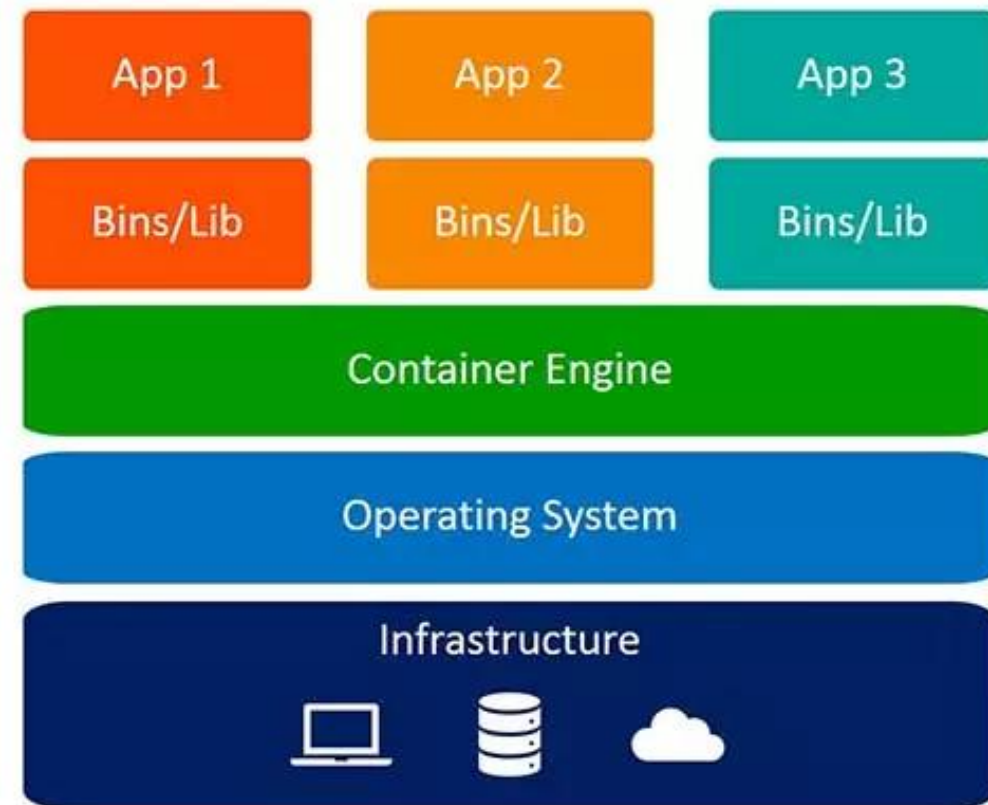
Tipo 2

Contenedores

La principal alternativa a la virtualización basada en hipervisores es la **contenedorización**. La virtualización del sistema operativo, por ejemplo, es un método de virtualización del kernel basado en contenedores. La virtualización del sistema operativo es similar a la partición. En esta arquitectura, un sistema operativo se adapta para que funcione como múltiples sistemas discretos, lo que permite implementar y ejecutar aplicaciones distribuidas sin lanzar una VM completa para cada una. En cambio, varios sistemas aislados, llamados contenedores, se ejecutan en un solo host de control y todos acceden a un solo kernel.



Virtual Machines



Containers