

# Sistema informático

- **Hardware**
  - Procesador(controla el funcionamiento del computador, realiza procesamiento, y se denomina unidad central de proceso *CPU*) y memoria
  - Dispositivos de E/S (entrada y salida): almacenamiento, red, *HCI*(Human Computer Interaction), impresión.
- **Software**
  - Software de sistema: compilador, *GUI*(interfaz gráfica de usuario), *shell*(intérprete de comandos), etc.
  - Aplicaciones
- **Personas**
  - Usuarios en general
  - Administradores del sistema
  - Desarrolladores/programadores

## Computadora: Elementos básicos

- Procesador (CPU).
- Memoria principal: Almacena datos y programas. es volátil; cuando se apaga el computador, se pierde lo almacenado. En contraste, lo guardado de la *memoria del disco* se mantiene incluso cuando se apaga el computador. se le denomina también memoria real o memoria primaria.
- Módulos de E/S: Transfieren los datos entre el computador y su entorno externo. está formado por diversos dispositivos, incluyendo dispositivos de memoria secundaria (por ejemplo, discos), equipos de comunicaciones y terminales.
- Bus del sistema: Proporciona comunicación entre procesadores, la memoria principal y los módulos de E/S

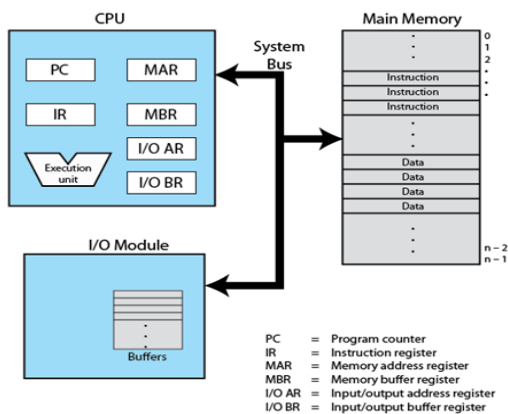


Figure 1.1 Computer Components: Top-Level View

## ¿Qué es un Sistema Operativo?

es un programa que controla la ejecución de aplicaciones y programas y que actúa como interfaz entre las aplicaciones y el hardware de la computadora

- Un sistema operativo es un programa que administra el hardware de una computadora.(**Objetivos:** Eficiente(permite que los recursos de un sistema de  
● computación se puedan utilizar de una manera eficiente), fácil de usar,  
evolutivo(se debe construir de tal forma que se puedan  
● desarrollar, probar e introducir nuevas funciones en el sistema sin  
● interferir con su servicio.)).

- Interactúa entre el/los usuario/s y el hardware.
- Un sistema informático tiene recursos que pueden ser útiles para solucionar un problema.
- El sistema operativo actúa como intermediario entre estos recursos: CPU, espacio de memoria, almacenamiento de archivos, dispositivos de E/S, etc.

## Objetivos y funciones primarias:

Abstracción: El SO se encarga de proporcionar abstracciones para que los programadores puedan enfocarse en resolver las necesidades particulares de los usuarios.

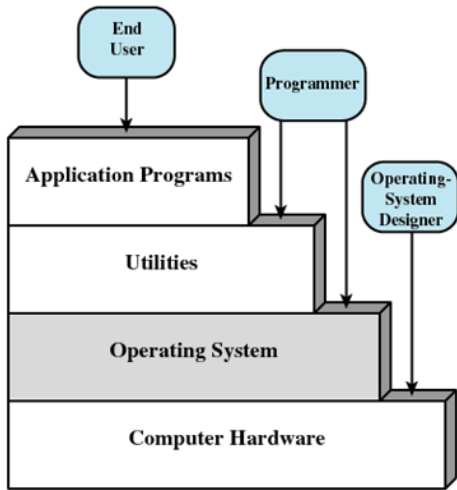
Administración de recursos: Una computadora tiene una gran cantidad de recursos. Al gestionar la asignación de recursos, el SO implementa políticas que los asigne de forma efectiva y acorde a las necesidades establecidas.

Aislamiento: En un sistema *multiusuario* y *multitarea* cada *proceso* y cada usuario no tendrá que preocuparse por otros que estén usando el mismo sistema. Su experiencia será la misma que si el sistema estuviera exclusivamente dedicado a su atención. Para implementar correctamente las funciones de aislamiento, el SO utiliza hardware específico para dicha protección.

### SO Interfaces:

// Preguntar al profe si esta tabla está correcta haciendo referencia a si las API fueron creadas para los desarrolladores(?)

|  |                           |
|--|---------------------------|
| Usuarios en general - Gráfica (GUI: Graphical User Interface)  | Entorno de ejecución      |
| Administradores - Texto (CLI: Command Line Interface) lenguaje sencillo para dar instrucciones al SO. Carga programas, archivos. Se le llama consola o shell o interprete de ordenes | Entorno de administración |
| Desarrolladores - Servicios de programación ( <b>API: Application Programming Interface</b> )  | Interfaz de programación  |



### Servicios proporcionados por un Sistema Operativo

Desarrollo de programas: proporciona utilidades y servicios, como editores y depuradores, para el programador.

Ejecución de programas: Las instrucciones y los datos se deben cargar en memoria principal. Los dispositivos de E/S y archivos se deben inicializar, y otros recursos deben prepararse. Los SO realizan estas tareas de planificación en nombre del usuario.

Acceso a dispositivos de E/S: Cada dispositivo de E/S requiere su propio conjunto de instrucciones o señales de control para cada operación(drivers). El SO proporciona una interfaz que esconde esos detalles de forma que los programadores puedan acceder a dichos dispositivos utilizando lecturas y escrituras sencillas.

Acceso controlado a los archivos: el SO debe conocer la naturaleza del dispositivo de E/S (disco, cinta, teclado, etc), y los datos contenidos en los archivos. En un sistema con múltiples

usuarios, el SO proporciona mecanismos de protección para controlar el acceso a los archivos(entre usuarios).

Acceso al sistema: Protección a recursos y datos. Evita el uso de usuarios no autorizados(inicio de sesión con usuario y contraseña) y resuelve conflictos en el uso de recursos.

Detección y respuesta a errores: Durante la ejecución de un programa pueden ocurrir *errores de hardware* internos y externos: error de memoria, fallo en un dispositivo; y *errores software*: división por cero, el intento de acceso a una posición de memoria prohibida o la incapacidad del SO para conceder la solicitud de una aplicación. El SO *debe* proporcionar una respuesta que elimine la condición de error, con el menor impacto en las aplicaciones que están en ejecución.

Auditoría: Un buen sistema operativo recoge estadísticas de uso de los diferentes recursos y supervisa parámetros de rendimiento tales como el tiempo de respuesta. Esta información es útil para anticipar las necesidades de mejoras y optimizar el sistema a fin de mejorar su rendimiento.

### SO como Administrador de recursos

|  |   |
|--|---|
| Un proceso es un programa en ejecución | El SO debe determinar a quién se le entregan los recursos, qué cantidad, en qué momento y por cuánto tiempo |
|--|---|

|   |   |
|---|---|
| Un <i>recurso</i> es un dispositivo físico o virtual que necesita un proceso para ejecutarse  | Una computadora es un conjunto de <i>recursos</i> que se utilizan para el transporte, almacenamiento y procesamiento de los datos |
| Los recursos son escasos y los procesos compiten por ellos  | Una porción del SO se encuentra en la memoria principal. Esto incluye el kernel (núcleo)  |
| SO es responsable de: <i>Crear y borrar – Suspender y reanudar procesos. sincronización; comunicación; tratamiento de interbloqueos de los procesos</i> | el sistema operativo debe determinar cuánto tiempo de procesador debe asignarse a la ejecución de un programa                     |

### SO como administrador de memoria

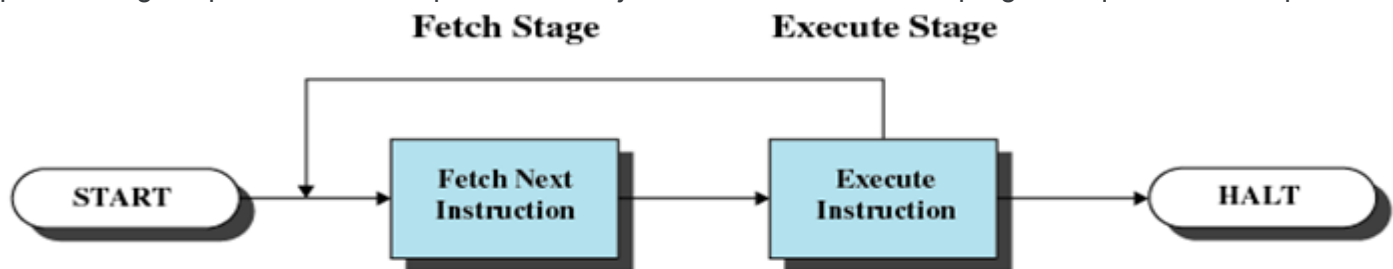
Para que un programa se ejecute debe estar en memoria principal (MP) mientras pasa accede a instrucciones y datos de la memoria, cuando termina la memoria queda disponible.

### Ejecución de instrucciones

1. El procesador lee (busca) instrucciones de la memoria, una cada vez, y ejecuta cada una de ellas. La ejecución del programa consiste en repetir el proceso de búsqueda y ejecución de instrucciones.
2. La ejecución de la instrucción puede involucrar varias operaciones dependiendo de la naturaleza de la misma.

**Se denomina ciclo de instrucción al procesamiento requerido por una única instrucción**

**Fase de búsqueda y fase de ejecución:** La ejecución del programa se detiene sólo si se apaga la máquina, se produce algún tipo de error irreparable o se ejecuta una instrucción del programa que detiene el procesador.



### Interrupciones

- Una interrupción suspende la secuencia normal de ejecución(memoria y E/S) pueden interrumpir la secuencia normal del procesado (manera de mejorar la utilización del procesador)
  - Cuando se completa el procesamiento de la interrupción, se reanuda la ejecución
  - El programa de usuario no tiene que contener ningún código para las interrupciones; el procesador y el SO son responsables de suspender el programa de usuario y, posteriormente, reanudarlo en el mismo punto

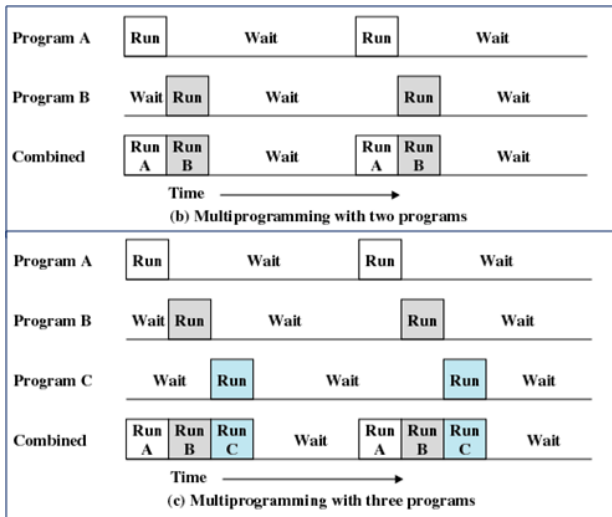
### Ciclo de instrucción con interrupciones

- El procesador comprueba si hay interrupciones
- Si no hay ninguna solicitud de interrupción, toma la siguiente instrucción para el programa actual
- Si una interrupción está pendiente, suspende la ejecución del programa actual y se ejecuta la rutina del manejador de interrupciones.

**Interrupciones Múltiples:** deshabilitan mientras se procesa un pedido de interrupción; Definen prioridades para la atención de pedidos; Secuencia temporal con múltiples interrupciones anidadas.

## Clases de Interrupciones:

- Programa: Generada por alguna condición que se produce como resultado de la ejecución de una instrucción, tales como desborde aritmético, una división por cero, un intento de ejecutar una instrucción de máquina ilegal, y las referencias fuera del espacio de la memoria permitido para un usuario.
- Temporizador: Generada por un temporizador del procesador. Permite al sistema operativo realizar ciertas funciones de forma regular.
- E/S: Generada por un controlador de E/S para señalar la conclusión normal de una operación o para indicar diversas condiciones de error
- Fallo de hardware: Generada por un fallo, como un fallo en el suministro de
  - energía o un error de paridad en la memoria.
  -



**Mono-programación:** El procesador debe esperar a que se complete la instrucción de E/S antes de continuar

**Multi-programación:** Llamado multitarea (multitasking) Los procesos entrelazan su ejecución: concurrencia. La CPU y la E/S trabajan a la misma vez – Se terminan más trabajos en menos tiempo

Cuando una tarea necesita esperar por la terminación de una operación de E/S, el procesador puede cambiar a otra tarea. (posible por las interrupciones).

Incluso con interrupciones, puede suceder que el procesador siga sin utilizarse eficientemente, cuando el tiempo de completar una operación E/S del usuario es mayor, el procesador estará parado gran parte del tiempo.

\*Una solución a este problema es permitir que múltiples programas de usuario estén activos al mismo tiempo. La CPU y la E/S trabajan a la misma vez (se terminan más trabajos en menos tiempo).

## Entornos de computación

- Computadoras personales
- Dispositivos de mano (mobile computing): móviles y tabletas
- Sistemas embebidos (embedded systems)
- Servidores + multiprocesadores
- Clusters de servidores
- Sistemas distribuidos, virtualizados, en la nube (cloud systems)

## Algunos tipos de Sistemas Operativos

- **Procesamiento por lotes (batch processing)** : Primera generación de SO (principios de los 1950) Objetivo: Automatizar la ejecución de trabajos y aumentar la utilización y ocupación del procesador. Los trabajos se agrupaban en lotes que se iban ejecutando en secuencia. Los más primitivos eran secuenciales; la multiprogramación se incorporó a finales de los '5. Primer lenguaje para dictar tareas al SO: JCL (Job Control Language)
- **Tiempo compartido (time sharing):** Sistemas de tiempo compartido. Cada proceso dispone de una pequeña rodaja (slice) de tiempo periódica. Los procesos se van turnando en el uso de la CPU, si la rodaja de tiempo es bastante pequeña (milisegundos), el usuario no percibe las pausas periódicas de su sesión.
- **Tiempo real (real time):** Diseñados para cumplir tareas que deben completarse en un plazo prefijado (sistemas de control industrial, sistemas multimedia, ...) Usan algoritmos especiales para la planificación del procesador. STR crítico
  - –Para industria y sistemas embebidos en los que el cumplimiento de plazos es crítico.
  - –Suelen prescindir de servicios que afectan a los tiempos (ejemplo: memoria virtual).
- **Sistemas multiusuarios:** Un sistema multiusuario reconoce que hay varios perfiles de acceso, con privilegios distintos:
  - –Permisos de acceso a archivos y aplicaciones.

- –Cuotas de espacio o de tiempo de procesador.
- –Prioridad en el acceso a los recursos.
- **Multiusuario ≠ multitarea:** Puede haber sistemas multitarea que no son multiusuario.
- **Máquinas virtuales:** En un sistema virtualizado los servidores centrales (uno o más) soportan la carga de trabajo de todos los usuarios. Los servidores tienen un software específico que permiten tener en ejecución, en la misma máquina, varias instancias del mismo sistema operativo, o incluso de diferentes. Para utilizar estos sistemas centralizados, se necesitan terminales, denominados ThinClients o terminales ligeros, que se encargan de mostrar la interfaz de los sistemas operativos remotos.
  - –Los terminales se conectan a los servidores centrales, quienes están ejecutando los sistemas operativos demandados y las aplicaciones de cada cliente.
  - –Los terminales son simples consolas para poder visualizar la interfaz del sistema operativo que corre en remoto.

Elementos: Drivers(maneja los periféricos) , Núcleo o Kernel(), Intérprete de comandos(shell), Proceso(es un programa en ejecución). Archivo(), Directorio(), Dispositivo virtual(). Discos (ejecutables o código binario). Usuario y grupo....

administrador de recursos

administrador de procesos

### Sistemas multiprocesadores

Varios modelos de acceso a memoria:

- UMA (Uniform Memory Access) La memoria física es compartida uniformemente por todo los procesadores.
- Los procesadores tipo NUMA (Non Uniform Memory Access) tienen tiempos de acceso a la memoria compartida que dependen de la ubicación del proceso y la memoria.

Varios modelos de ejecución de procesos:

- SMP multiprocesamiento simétrico: una tarea se puede ejecutar en cualquier procesador.
- AMP multiprocesamiento asimétrico: hay especialización de tareas (ejemplo: un procesador ejecuta el SO y otro los procesos de usuario)
- \*sistemas que tienen más de un procesador

### Sistemas distribuidos

Un sistema distribuido es un conjunto de computadoras conectadas en red y que se utiliza como si fuera un único sistema con múltiples procesadores + una gran memoria compartida + un gran almacenamiento secundario.

No existe un “sistema distribuido universal”, pero sí hay servicios con características de sistema distribuido

- Servicios en la nube (Dropbox, Amazon, ...)
- World Wide Web.

## UNIDAD DOS

# GESTIÓN DE PROCESOS

- **Identificador:** Identificador único, para distinguirlo del resto de procesos
- **Estado:** Indica el proceso actual del proceso
- **Prioridad:** Nivel de prioridad relativo al resto de procesos
- **Contador de programa:** La dirección de la siguiente instrucción del programa que se ejecutará
- **Punteros a memoria:** Incluye los punteros al código de programa y los datos asociados, y de bloques de memoria compartidos con otros procesos
- **Datos de contexto:** Los datos que están presentes en los registros del procesador cuando el proceso está corriendo
- **Información de estado de E/S:** Incluye las peticiones de E/S pendientes, dispositivos de E/S asignados a dicho proceso, lista de archivos en uso, etc.

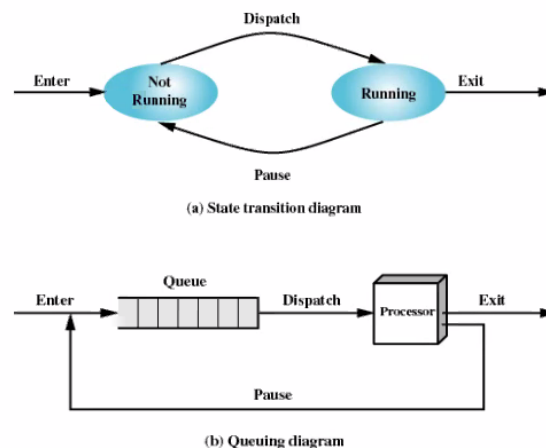
- **Información de auditoría:** Puede incluir la cantidad de tiempo de procesador y de tiempo de reloj utilizados, así como los límites de tiempo, registros contables, etc.

## Bloque de control de proceso

- Cada proceso debe tener una estructura de datos que almacena su estado e información para su control
- Contiene los elementos del proceso
- El BCP es creado y gestionado por el sistema operativo
- Permite soporte para múltiples procesos

## Modelo de proceso de 2 estados

- Existe una cola cuyas entradas son punteros al BCP de un proceso en particular
- Un proceso que se interrumpe se transfiere a la cola de procesos en espera
- Si el proceso ha finalizado o ha sido abortado, se descarta (sale del sistema)
- El activador selecciona un proceso de la cola para ejecutar



## Modelos de 2 estados. Limitaciones e inconvenientes

- La gestión de colas mencionada, sería efectiva si todos los procesos estuviesen siempre preparados para ejecutar
  - La cola es una lista de tipo FIFO y el procesador opera siguiendo una estrategia cíclica (*round-robin* o turno rotatorio) sobre todos los procesos disponibles (cada proceso de la cola tiene cierta cantidad de tiempo, por turnos, para ejecutar y regresar de nuevo a la cola, a menos que se bloquee)
- Sin embargo, esta implementación es inadecuada
  - Algunos procesos que están en el estado No Ejecutando están listos para ejecutar, mientras que otros están bloqueados, esperando a que se complete una operación de E/S
  - Por lo tanto, utilizando una única cola, el activador debería recorrer la lista buscando los procesos que no estén bloqueados y que lleven en la cola más tiempo
- Una forma más natural para manejar esta situación es dividir el estado de No Ejecutando en dos estados: Listo y Bloqueado
- Para hacer esta nueva gestión correctamente, se añaden dos estados adicionales (Nuevo y Terminado) que resultan muy útiles

## Modelo de 5 estados

Los estados **Nuevo (New)** y **Saliente (Exit)** son útiles para construir la gestión de procesos.

**Nuevo** se corresponde con un proceso que acaba de ser definido, Asocia un identificador a dicho proceso, reserva y construye las tablas necesarias para gestionar al proceso.

un proceso sale del sistema en dos fases: Primero, el proceso termina cuando alcanza su punto de finalización, es abortado debido a un error, o que otro proceso con autoridad causa que se aborte. Luego, la



terminación mueve el proceso al estado **saliente**. El proceso no es elegible de nuevo para su ejecución. Las tablas e información asociada se encuentran temporalmente preservadas por el sistema operativo, el cual proporciona tiempo para que programas auxiliares o de soporte extraigan la información necesaria. Una vez que estos programas han extraído la información necesaria, el sistema operativo no necesita mantener ningún dato relativo al proceso y el proceso se borra del sistema.

## UNIDAD 3

planificación  
son las acciones que da el cpu para los procesos  
--

## UNIDAD 4 GESTIÓN DE LA MEMORIA

LA MEMORIA ESTÁ DIVIDIDA EN USUARIO Y SISTEMA  
LA PARTE DE USUARIO ESTÁ DIVIDIDA EN PORCIONES O PARTES  
puntos críticos  
espacios de direcciones  
Direccionamiento  
intercambio  
gestión de la memoria requisitos  
reubicación  
protección  
direccion logica  
fragmentación externa

---

## UNIDAD 5

### Asignación de direcciones IP

Dependiendo de la implementación concreta, el servidor DHCP tiene tres métodos para asignar las direcciones IP:

- manualmente, cuando el servidor tiene a su disposición una tabla que empareja **direcciones MAC** con direcciones IP, creada manualmente por el administrador de la red. Sólo los clientes con una dirección MAC válida recibirán una dirección IP del servidor.
- dinámicamente, el único método que permite la reutilización de direcciones IP. El administrador de la red asigna un intervalo de direcciones IP para el DHCP y cada ordenador cliente de la LAN tiene su software de comunicación TCP/IP configurado para solicitar una dirección IP del servidor DHCP cuando su tarjeta de interfaz de red se inicie. El proceso es transparente para el usuario y tiene un periodo de validez limitado.
- automática, asigna una dirección IP a una máquina cliente cuando esta hace por primera vez una solicitud al servidor DHCP, y hasta que el cliente la libera.  
Se suele utilizar cuando el número de clientes no varía demasiado.

A veces se presta a confusión el **router (enrutador)** con **puerta de enlace**: la puerta de enlace es en definitiva la dirección IP del router. Dirección que ha de estar dentro de la subred. La dirección IP del router se programa en el mismo router. La mayoría de los router vienen con una dirección de fábrica, modificable a través de un puerto serie o por red mediante http, u otros protocolos. Esta dirección modificable es la puerta de enlace de la red.

Rango de Direcciones IP disponibles por Clases de Red A, B, C, D, E

**Clase A:** 0.0.0.0 a 127.255.255.255 – Máscara /8

Reservado **0.0.0.0/8** (de 0.0.0.0 a 0.255.255.255)

Reservado **127.0.0.0/8** (de 127.0.0.0 a 127.255.255.255)

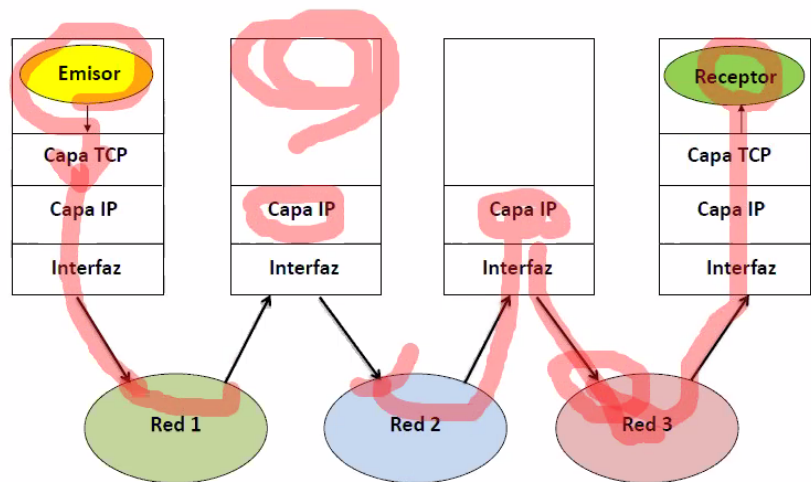
**Clase B:** 128.0.0.0 a 191.255.255.255 – Máscara /16

**Clase C:** 192.0.0.0 a 223.255.255.255 – Máscara /24

**Clase D:** 224.0.0.0 a 239.255.255.255 (Clase reservada para **Múlticas**)

**Clase E:** 240.0.0.0 a 255.255.255.255 (Clase reservada)

| Clase | Rango válido del Primer Octeto | Redes válidas               |
|-------|--------------------------------|-----------------------------|
| A     | 1 a 126                        | 1.0.0.0 a 126.0.0.0         |
| B     | 128 a 191                      | 128.0.0.0 a 191.255.255.255 |



funciones de las capas  
por qué existen las capas:

04/05/2021

TRABAJO AUTÓNOMO

en equipo, es una investigación sobre DEVOPs es otra metodología ágil para organizar la empresa herramienta vmware

la automatización es un proceso que se hace para ayudar al desarrollo (o para el testing)

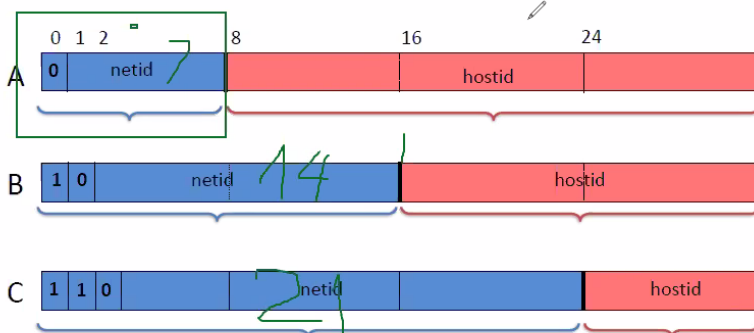
lo que hace QA son reportes de testing

el objetivo es conocer de forma conceptual características y cualidades tal y devops

es un conjunto de cosas que se pueden aplicar a diferentes áreas



## Clases de direcciones IP



## UNIDAD 7

## VIRTUALIZACIÓN Y MÁQUINAS VIRTUALES



**Virtualizar:** (software que permite simular una computadora dentro de otra) Virtualización es una tecnología que permite que S-O se ejecuten como aplicaciones dentro de otros S-O.

La virtualización es un software que incluye emulación. La emulación se utiliza cuando el tipo de CPU de origen es diferente del tipo de CPU de destino. este concepto se puede extender para la virtualización de SO

*anfitrión/host:* equipo físico sobre el que se simula otros equipos.

*invitado/guest:* equipo simulado sobre el sistema real

**-Software de emulación:** cada instrucción a nivel de máquina que se ejecuta de forma nativa en el sistema de origen debe traducirse a la función equivalente en el sistema de destino.

La idea de una máquina virtual es abstraer el hardware de la computadora, formando entornos de ejecución diferentes, creando así la ilusión de que cada entorno de ejecución tiene su propia computadora.

Una **MV(VM)** es un software capaz de albergar otro sistema operativo haciendo parecer que se tiene una computadora real. "el Sistema operativo que instalamos en un mv es el huésped"

## HIPERVISORES

software que aprovecha y gestiona los recursos del sistema anfitrión para crear MV (es un programa que se descarga)

-este software de virtualización implementa **HYPERVERISOR** o **VMM(Virtual Machine Monitor)**; en una capa de abstracción entre el hardware de la máquina física(*host*, anfitrión) y la MV formada por el hardware y software virtualizado, haciendo el papel de intermediario entre lo real y lo virtualización.

-La capa de software hipervisor, gestiona los cuatro recursos principales de una computadora: CPU, memoria, almacenamiento y conexiones de red, repartiendo estos recursos entre las MV creadas en la PC anfitriona, esto permite tener varias MV ejecutándose en la misma.

## HIPERVISOR 1

"virtualización en modo nativo", este software se instala directamente sobre el equipo haciendo funciones de (S-O) como de virtualización.

Este método se utiliza generalmente en las empresas que pueden disponer de uno o varios servidores dedicados a la virtualización de sistemas.

## HIPERVISOR TIPO 2

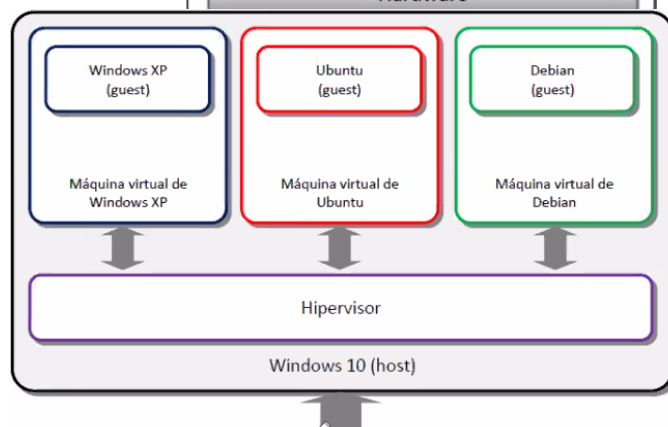
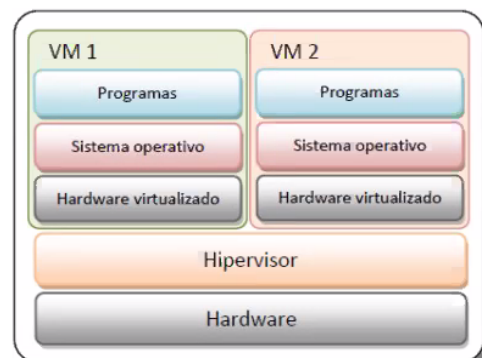
El software alojado (*hosted*), se caracteriza por ser instalado en un equipo que cuente con un SO previo anfitrión (como Linux, Microsoft Windows, Mac OS X). el método de virtualización apropiado e ideal para probar software (SO y app) sin riesgo de afectar nada que haya en la máquina anfitriona. Ésta forma de virtualización es menos eficiente que la anterior, pero se puede seguir utilizando el equipo físico con otras aplicaciones en el SO anfitrión.

## Máquina virtual de sistema

una máquina virtual de sistemas es aquella que emula a una computadora completa.

-el software puede hacerse pasar por otro dispositivo -como una PC- de modo que se ejecuta otro SO en su interior. Tiene su propio disco duro, memoria, tarjeta de video y demás componentes de hardware, aunque todos ellos son virtuales.

-que sus componentes sean virtuales no significa que necesariamente no existan. Por ejemplo, una máquina virtual puede tener recursos reservados de 1GB de RAM y 80 GB de



almacenamiento, que salen de algún lugar de la computadora donde está instalada la MV (host). otros dispositivos podrían realmente ser inexistentes físicamente, como por ejemplo un CD ROM que puede ser el contenido de una imagen de ISO en vez del leído en un lector de CD verdadero. “Una máquina virtual no puede acceder al resto de datos de la máquina anfitrión”

### Máquinas virtuales de proceso

Una MV de proceso tiene menor alcance que una de sistema. En vez de emular una PC por completo, una MV de proceso ejecuta un proceso concreto, como una aplicación, en su entorno de ejecución.

**Beneficios del uso de máquinas virtuales** (brinda una protección completa de los diversos recursos del sistema)

Algunos de los principales usos de las máquinas virtuales de sistema son:

- Para probar sistemas operativos
- Para ejecutar programas antiguos
- Para usar aplicaciones disponibles para otros sistemas operativos
- Para probar una aplicación en distintos sistemas
- Seguridad adicional
- Para aprovechar su gran dinamismo

Todos estos usos tienen una contrapartida o efecto negativo: el **rendimiento**. si el hardware de la PC se usa para soportar dos (o más) sistemas operativos a la vez, en vez de uno, el rendimiento se resiente.

¿Cómo se usa una máquina virtual? lo primero es instalar una aplicación en la PC, que pueda crearla o al menos reproducirla. tal como VirtualBox, VMWare, QEMU, Parallels, etc.

### Operaciones con máquinas virtuales

La opción de **red interna** permite seleccionar o crear una que conecta dos o más máquinas virtuales. el equipo *host* no podrá ver el tráfico en la red virtual. La creación de una red virtual permite probar configuraciones y opciones de seguridad sin perder ancho de banda, estudiar la configuración defensiva contra virus, sin correr el riesgo de infectar otros sistemas

### Red entre Máquinas Virtuales y Físicas

Requisitos:

- Tener instalado el VirtualBox y una máquina virtual.
- Tener instaladas las Guest Additions para el sistema huésped.
- Los equipos deben estar conectados a un *router*.

**esta facil parece...**

### Escritorios remotos

Se utilizan para realizar una conexión remota de una PC a otra. Existen dos formas básicas de realizar una conexión remota:

**Por Sistema Operativo:** Cuando se realiza por S-O hay que configurar parámetros dependiendo del S-O. Generalmente se habilita la opción de acceso remoto y se dan permisos a los usuarios que van a tener acceso a la computadora.

**Por software de terceros:** Cuando se realiza por software de terceros es más fácil ya que se tiene un cliente y un servidor y lo único que se necesita es la dirección IP y el software se encarga del resto.

## Tipos de configuraciones de red en Virtual Box

Virtual Box: Una máquina virtual puede tener hasta 8 tarjetas de red Ethernet. El hardware de la tarjeta de red se elige en la sección de Red de la MV, en el apartado de Avanzadas denominado Tipo de adaptador.

**Modo No conectado:** VirtualBox informa al usuario de VM que existe una tarjeta de red, pero no hay conexión, Establecer el modo *No conectado* y cambiar a otro modo de red, fuerza al SO a reconfigurar la tarjeta de red.

**Modo NAT:** modo adecuado para la conexión a Internet, descargar archivos, leer el correo electrónico, etc. VirtualBox coloca un *router* virtual entre el exterior y el SO invitado(están en redes aisladas). tiene un servidor DHCP que provee direcciones IP hacia el interior. Este *router* mapea el tráfico desde y hacia la MV de forma transparente.

**Modo Red NAT:** la red funciona como LAN(red de área local. grupo de computadoras y dispositivos periféricos que comparten una línea de comunicaciones común o un enlace inalámbrico a un servidor dentro de un área geográfica específica.), los equipos dentro de la red NAT podrán comunicarse entre sí, y hacia el exterior a través del *router* NAT.

**Modo adaptador puente:** simula que la tarjeta virtual está conectada al mismo *switch* que la tarjeta física del anfitrión. La MV se va a comportar como un equipo más dentro de la red física en la que está el equipo anfitrión.

**Modo red interna:** esta configuración de tarjetas de red en modo *red interna*, se pueden construir redes aisladas, en las cuales solo habrá comunicación entre las máquinas virtuales que conforman una misma red interna. Para conectar al exterior, habrá que crear varias máquinas virtuales que funcionen como *routers*

**Modo sólo anfitrión:** se utiliza para crear una red interna que pertenecerá también el equipo anfitrión.

**¿Qué es?** VirtualBox es una aplicación **que sirve para** hacer MV con instalaciones de SO. Esto quiere decir **que** si tienes un ordenador con Windows, GNU/Linux o incluso macOS, puedes crear una máquina virtual con cualquier otro sistema operativo **para** utilizarlo dentro del **que** estés usando

### Exportar una máquina virtual

Es un mecanismo directo para exportar máquinas virtuales a otros equipos e incluso a otro software de virtualización distinto. Se basa en el denominado *Open Virtualization Format (OVF)*, un formato abierto para la exportación de máquinas virtuales.

### Importar una máquina virtual

Permite importar una VM, contenida en un único archivo con extensión *.ova*, desde el administrador de VirtualBox.

### Clonar una máquina virtual

Consiste en hacer una nueva máquina virtual exactamente igual a la primera pero con otra identidad, ya que los discos virtuales van a tener números de identificación (*uuid*) diferentes.

De esta forma la máquina virtual original y la clonada se podrían ejecutar en la misma computadora.

Si se hubiese hecho una MV a partir de la copia del archivo del disco virtual de otra, no se podrán ejecutar las dos en la misma PC, debido a que habría dos discos con el mismo *uuid*.

La clonación un recurso muy utilizado porque permite hacer una copia de seguridad de una máquina virtual recién creada.



# Operaciones con máquinas virtuales-redes

Conectar múltiples máquinas virtuales permite crear redes virtuales para una variedad de propósitos.

La opción de red interna permite seleccionar o crear una red interna que conecta dos o más máquinas virtuales. Incluso el equipo host no podrá ver el tráfico en la red virtual.

La creación de una red virtual permite probar configuraciones y opciones de seguridad sin perder ancho de banda, estudiar la configuración defensiva contra virus, sin correr el riesgo de infectar otros sistemas.

El hardware de la tarjeta de red se elige en la sección de Red de la MV.

## MicroServicios

**Los microservicios son:** enfoque arquitectónico y organizativo para el desarrollo de software donde el software está compuesto por pequeños servicios independientes que se comunican a través de APIs bien definidas. ---Los microservicios hacen que las aplicaciones sean más fáciles de escalar y más rápidas de desarrollar.

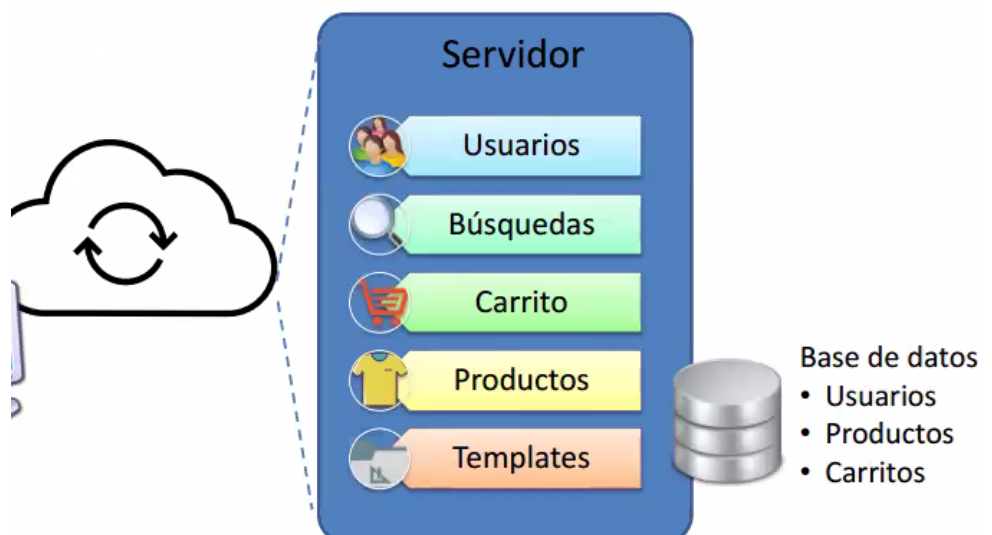
## Antecedentes: Sistema Monolítico

Un sistema monolítico es autónomo **no** depende de otras piezas de software y obedece únicamente a los componentes que están en su interior  
----->

“API” (Application Programming Interface) : son un conjunto de especificaciones y reglas/protocolos que permiten que un programa se comunique con otro.

-Si se trata de una llamada a sistema, se emplean APIs del Kernel de Linux.

\*Una API lleva a cabo sus funciones desde dentro de un programa de software. Si se trata de aplicaciones web, la API tiene su base en la web.\* \*Un servicio Web utilizará, la mayor parte de las veces, protocolo



HTTP para lograr la comunicación. También puede emplear SOAP, REST y XML-RPC\*

## las api con fundamentales en el desarrollo de software contemporáneo

**Rest “stateless”** → un servicio REST no tiene estado (es stateless), lo que quiere decir que, entre dos llamadas cualesquiera, el servicio pierde todos sus datos. ... De ahí el nombre: el estado lo mantiene el cliente y por lo tanto es el cliente quien debe pasar el estado en cada llamada

### Arquitecturas monolíticas:

Los procesos están asociados y se ejecutan como un solo servicio. Si uno experimenta demanda, se debe escalar toda la arquitectura. Agregar o mejorar características de una aplicación monolítica se vuelve complejo. Estas aumentan el riesgo de la disponibilidad de la aplicación porque muchos procesos dependientes y vinculados aumentan el impacto del error de un proceso.

### Arquitectura de microservicios:

se crea con componentes independientes que ejecutan procesos de la aplicación como un servicio. Estos se comunican a través de una interfaz definida mediante APIs. Los servicios se crean para que cada servicio desempeñe una sola función. Como se ejecutan de forma independiente, cada servicio se puede actualizar,

implementar y escalar para satisfacer la demanda de funciones específicas de la aplicación

### Características de los Microservicios:

**Autónomos:** se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios.

-No comparten ninguno de sus códigos. -Todas las comunicaciones ocurren a través de una API

**Especializados:** se enfoca en resolver un problema específico. -el servicio se vuelve complejo, se puede dividir en servicios más pequeños.

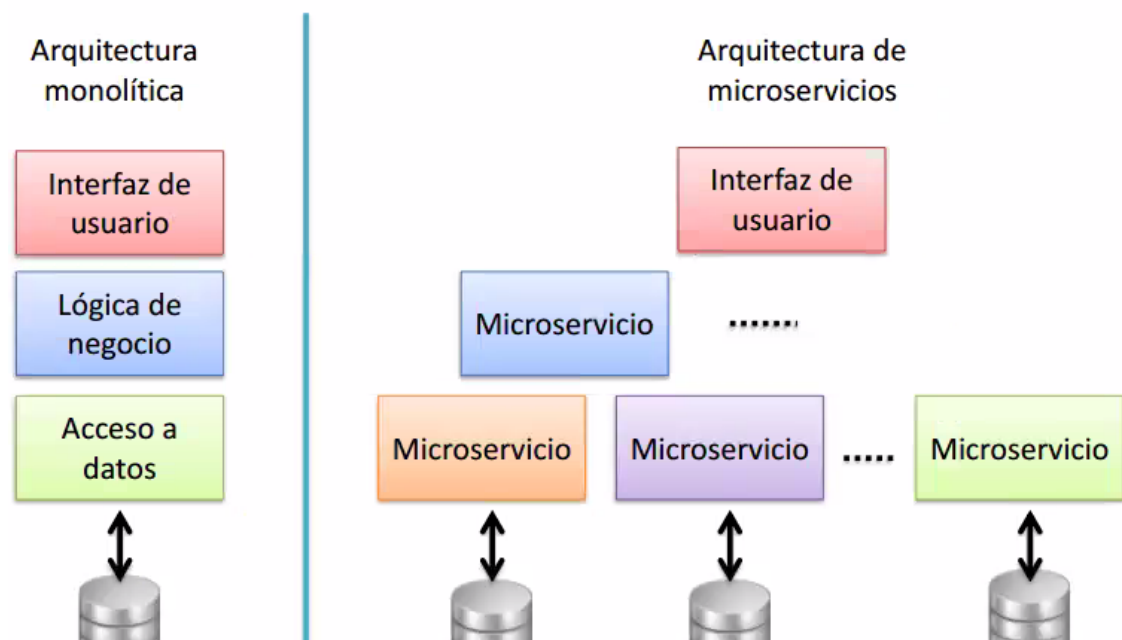
**Beneficios :** Agilidad; Escalado flexible; Implementación sencilla; Libertad tecnológica; Código reutilizable; Resistencia a errores; Resistencia a errores; Proyectos pequeños...

Desventajas : Alto consumo de recursos; Inversión de tiempo inicial; Complejidad en la gestión; Perfil de desarrollador; No uniformidad; Dificultad en la realización de pruebas; Coste de implementación alto

### Componentes :

- **Registro:** Proporciona independencia de la ubicación lógica o física.
- **Servidor perimetral:** Es el punto de acceso a cualquier llamada externa.
- **Balanceador :** Reparte la carga entre los servicios.
- **Servidor de configuración:** Proporciona la configuración del sistema.
- **Sistema de tolerancia a fallos:** Objetivo evitar fallos en cascada.
- **Gestión de logs:** Sistema para poder explotar la trazabilidad de los servicios

## Arquitecturas monolítica – microservicios



## **Conclusión**

La arquitectura de microservicios define una forma de trabajar con grandes aplicaciones en forma desacoplada. Surgen para atender demandas muy exigentes en cuanto a alta disponibilidad, capacidad de adaptación y cambio. No tiene por qué ser la mejor solución para todos los casos. En general, está indicado para grandes aplicaciones, con grandes volúmenes de peticiones (especialmente si se sufren fuertes picos de demanda) y con necesidad de alta disponibilidad. Si esta es la situación, esta arquitectura ayudará a ser más ágiles y conseguir atender las necesidades de una forma estructurada y probada.