

# INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

---

Karim Guevara Puente de la Vega  
2020

# Objetivos

- Proporcionar una visión general de los principales componentes de los Sistemas Operativos.
- Describir los principales conceptos involucrados.
- Reconocer la estructura de un Sistema Operativo.

# Agenda

- Introducción
- Arquitectura de un computador
- Computadoras y software
- Abstracción y compartición de recursos
- Qué es un sistema operativo?
- Estrategias de SOs

# Introducción

- Un SO actúa como un intermediario entre el usuario y el hardware
- El propósito de un SO es proporcionar un entorno en el que un usuario pueda ejecutar programas.



# Sistema Operativo

## □ SO es como un director

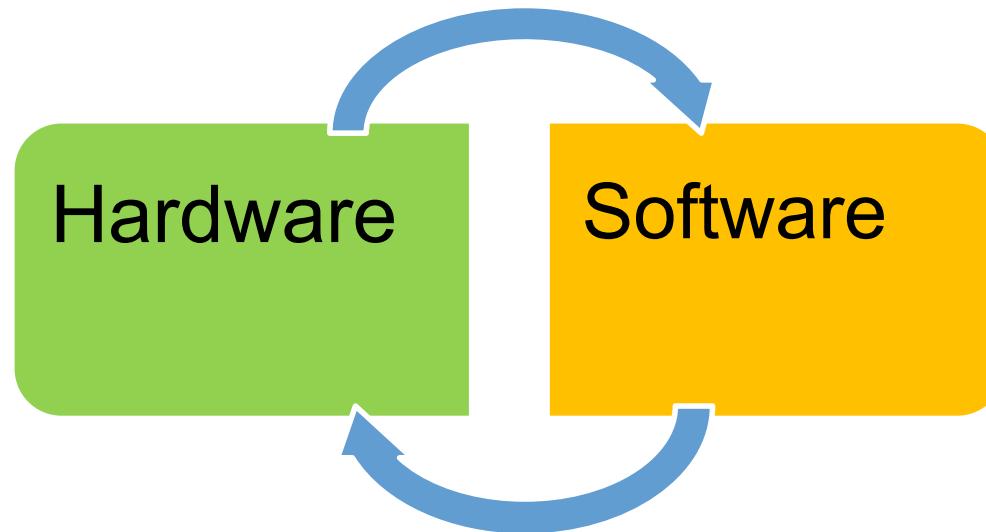


Un SO de altas prestaciones que proporciona poca funcionalidad fuerza a trabajar más sobre los programas de aplicación

- Asigna los recursos del computador a los diversos programas.
  - Sincroniza sus actividades individuales
  - Proporciona mecanismo para que los programas se ejecuten en perfecta armonía.
- 
- **Claves**
    - **Eficiencia** : fija un estandar de prestaciones
    - **Funcionalidad** : un SO proporciona un gran cantidad de funciones

# Computadores y Software

- La arquitectura de un computador está dividida en dos partes importantes:



# Computadoras y software

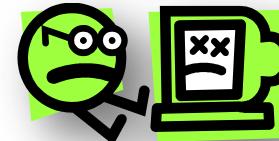
- El software se diversifica por su finalidad:
  - Software de aplicación
  - Software del sistema
    - **Sistema Operativo**



Corta imprime  
envía guarda



malloc() open()  
fork()

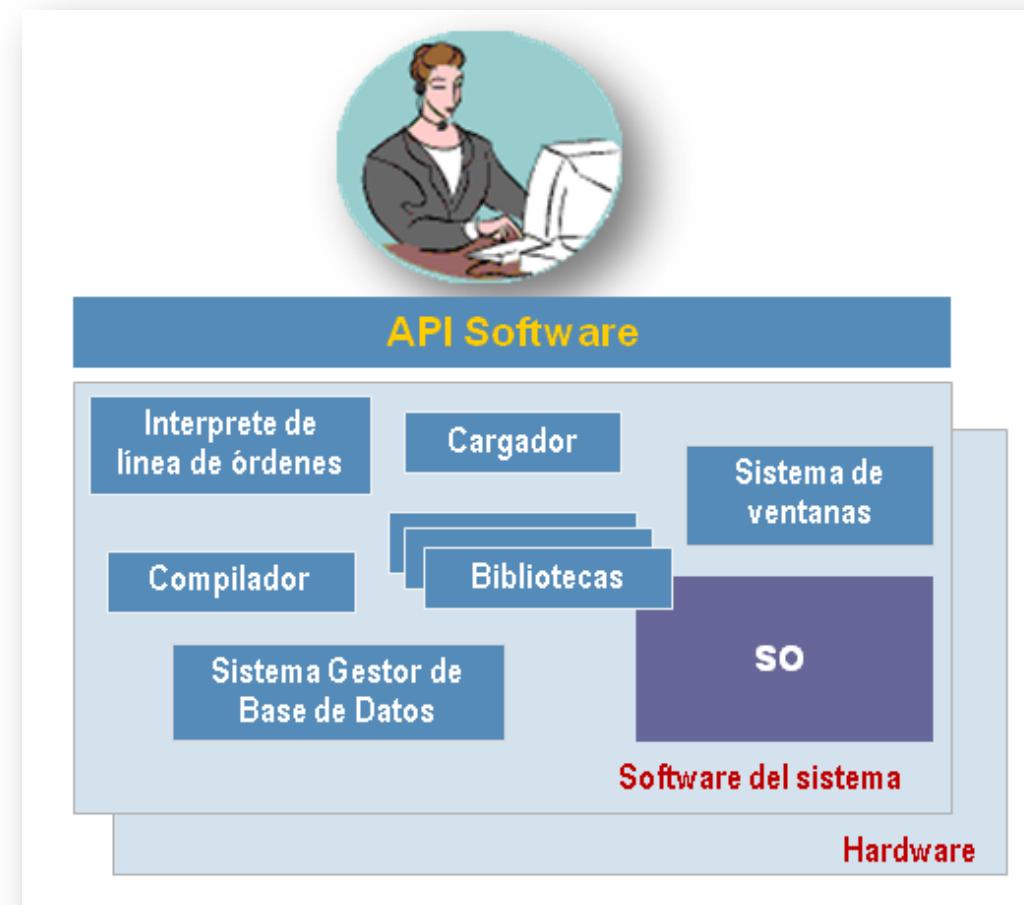


inicia-impresora  
lee-disco  
sigue-ratón

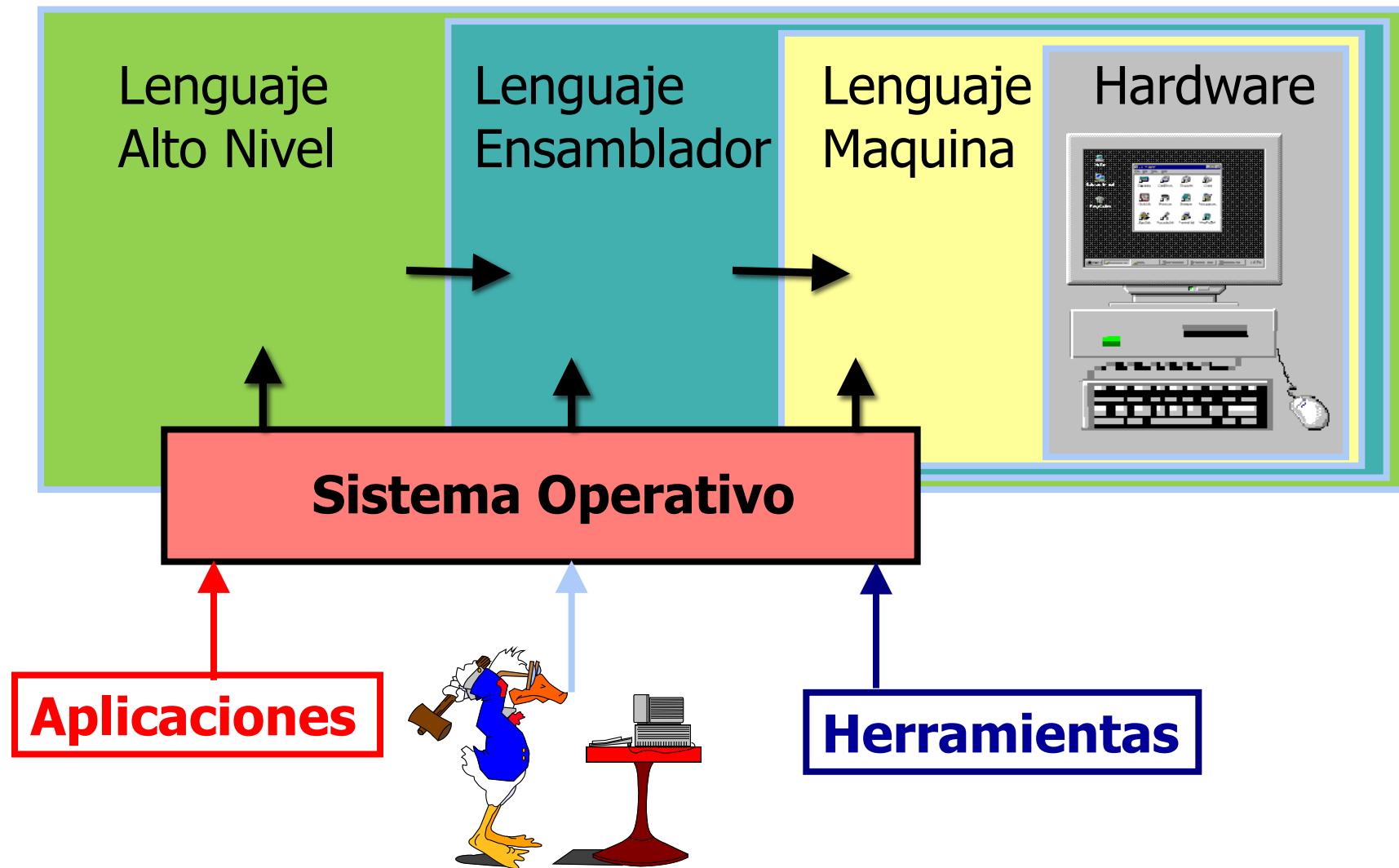


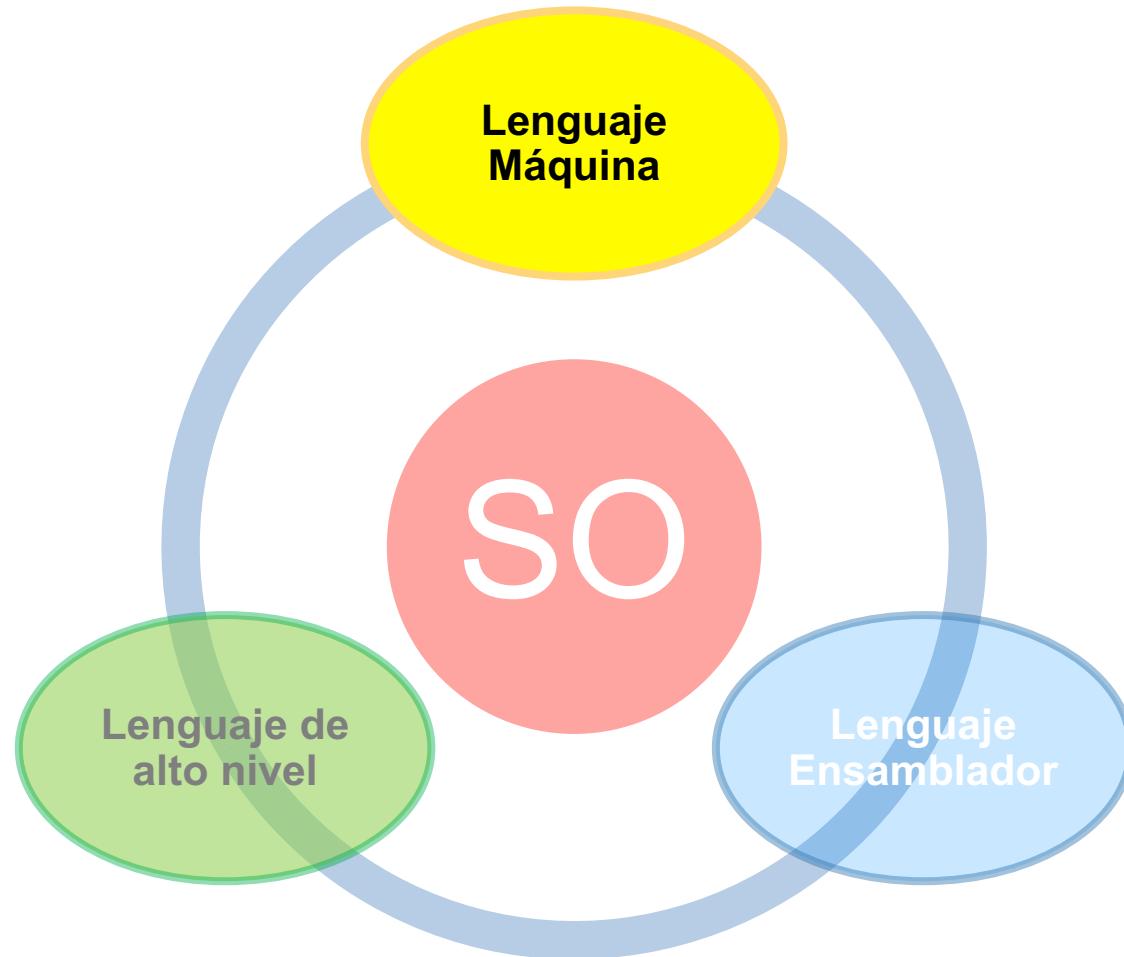
# Software del sistema general

- Interfaz de programación de aplicaciones (API)



# Arquitectura de Software



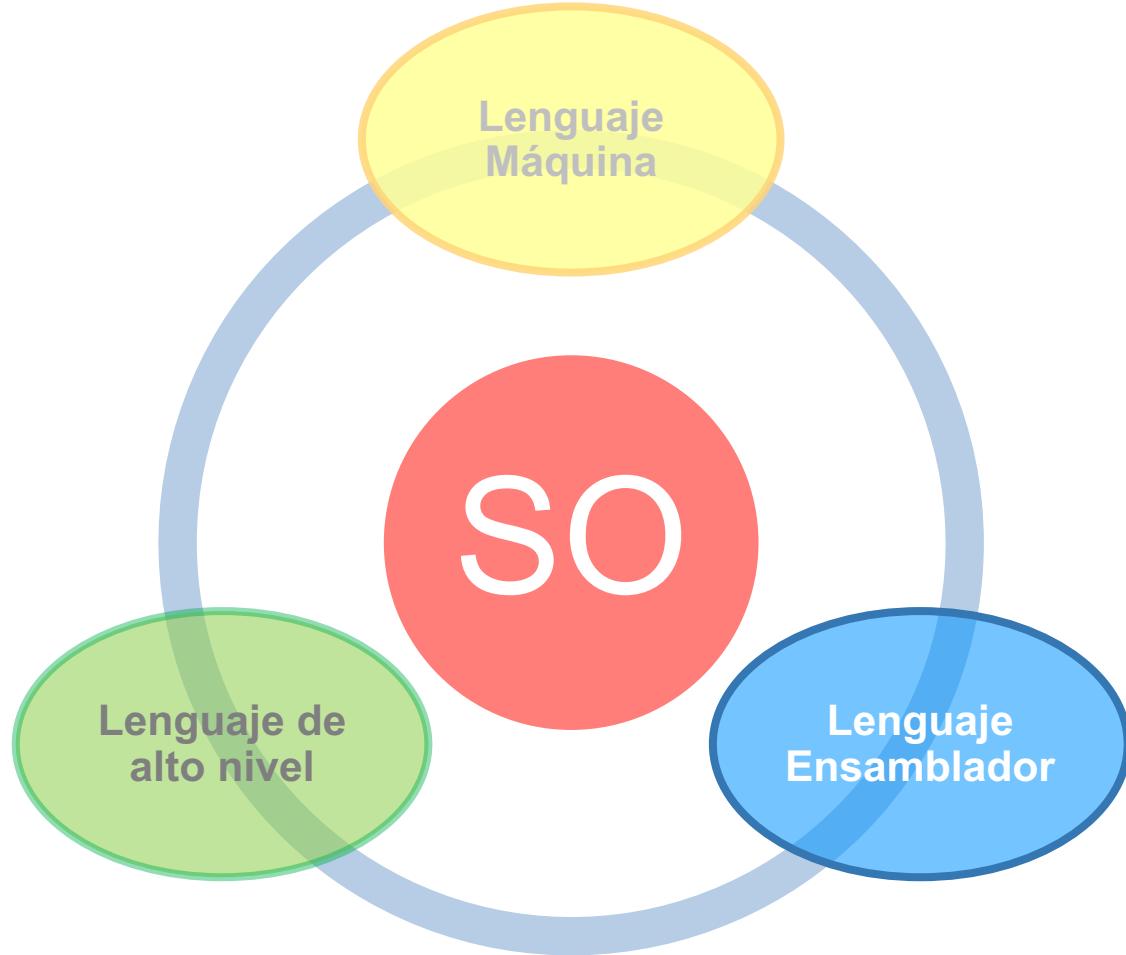


Único lenguaje que entiende el hardware

Específico para cada hardware  
(procesador, dispositivos, etc.).

Usa exclusivamente el sistema binario

Programa almacenado en memoria,  
contiene instrucciones y datos.

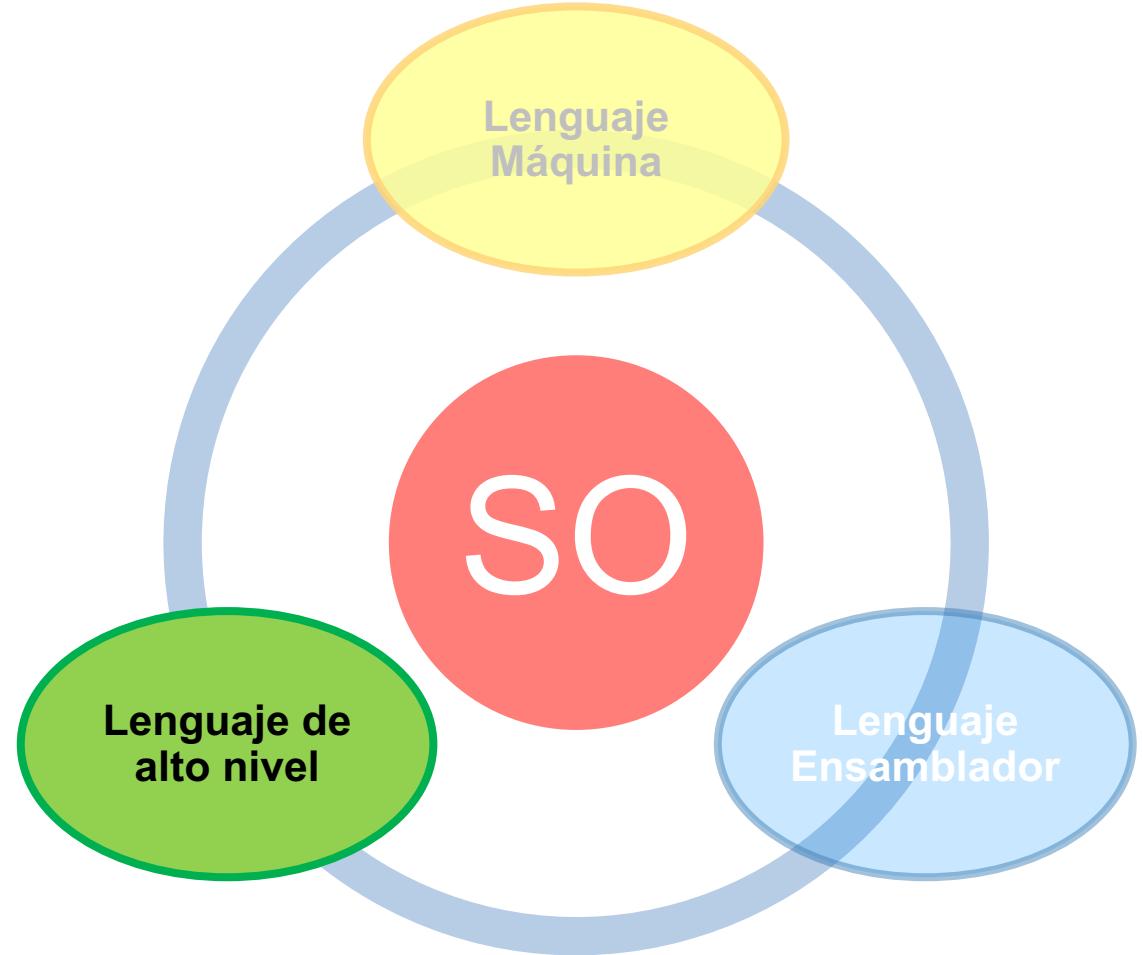


Usa mnemónicos para codificar las operaciones.

Es específico (aunque no único) para cada lenguaje de máquina.

Estructura de una instrucción:  
MNEMONICO ARGUMENTO(S)

Los datos y/o direcciones son codificados generalmente como números en sistema hexadecimal.



Basado en una estructura gramatical que permite el anidamiento de instrucciones.

Cuenta con un conjunto de palabras reservadas, para codificar estructuras de control y/o instrucciones.

Permite el uso de símbolos aritméticos y/o relacionales.

Permite al programador olvidarse del direccionamiento de memoria

En general, permite la realización de programas independiente del Hardware.

# ¿Cómo se distingue un SO del resto del sw del sistema?

- Un SO interactua directamente con el hardware para proveer una interfaz que se emplea en el resto de software del sistema y en el software de aplicación
- Un SO de propósito general es independiente del dominio
- El programa de aplicación usa recursos **abstractos** provistos por el SO para determinar su interacción concreta con los componentes hardware.
- El SO permite que diferentes aplicaciones **compartan** los recursos hardware en virtud de su política de gestión de los recursos.

# Abstracción de recursos

- El software del sistema provee un modelo abstracto de cómo funcionan los componentes del hardware.
  - Simplifica la forma como el programador de aplicaciones controla el hardware.
  - También limita la flexibilidad de cierto hardware concreto.
- Abstracciones de nivel más bajo están en el SO, las de nivel alto están en el software del sistema exterior al SO.
- Definir un conjunto de abstracciones que serán generalizadas para varios recursos, siendo intuitivas y adecuadas para más de un dominio.

# Compartición de recursos

- El SO conmuta el hardware entre los programas a una velocidad muy elevada:
  - Es aparente la ejecución simultanea de programas
- Computadores pueden soportar una auténtica operación simultánea en algunos casos.
- **Programas concurrentes y paralelos**, deben compartir el mismo computador.
  - Compartición transparente
  - Compartición explícita

# Máquinas abstractas y compartición transparente

- **Máquina abstracta:** es una simulación de un computador real.
  - **Proceso:** programa ejecutándose en una máquina abstracta.



- Compartición multiplexada en **espacio** y en **tiempo**
  - **Multiprogramación**

# Compartición explícita de recursos

- Permiten que los procesos usen recursos comunes con su propia estrategia
  - **Aislamiento de recursos:** obligación del SO de prevenir el acceso no autorizado a los recursos por una máquina abstracta, cuando están ya asignados.
    - P.e. aislamiento de la memoria, procesador
  - **Compartición autorizada** es deseable cuando los procesos deben cooperar entre sí, por tanto, deben de poder acceder al recurso compartido

# Qué es un Sistema Operativo?

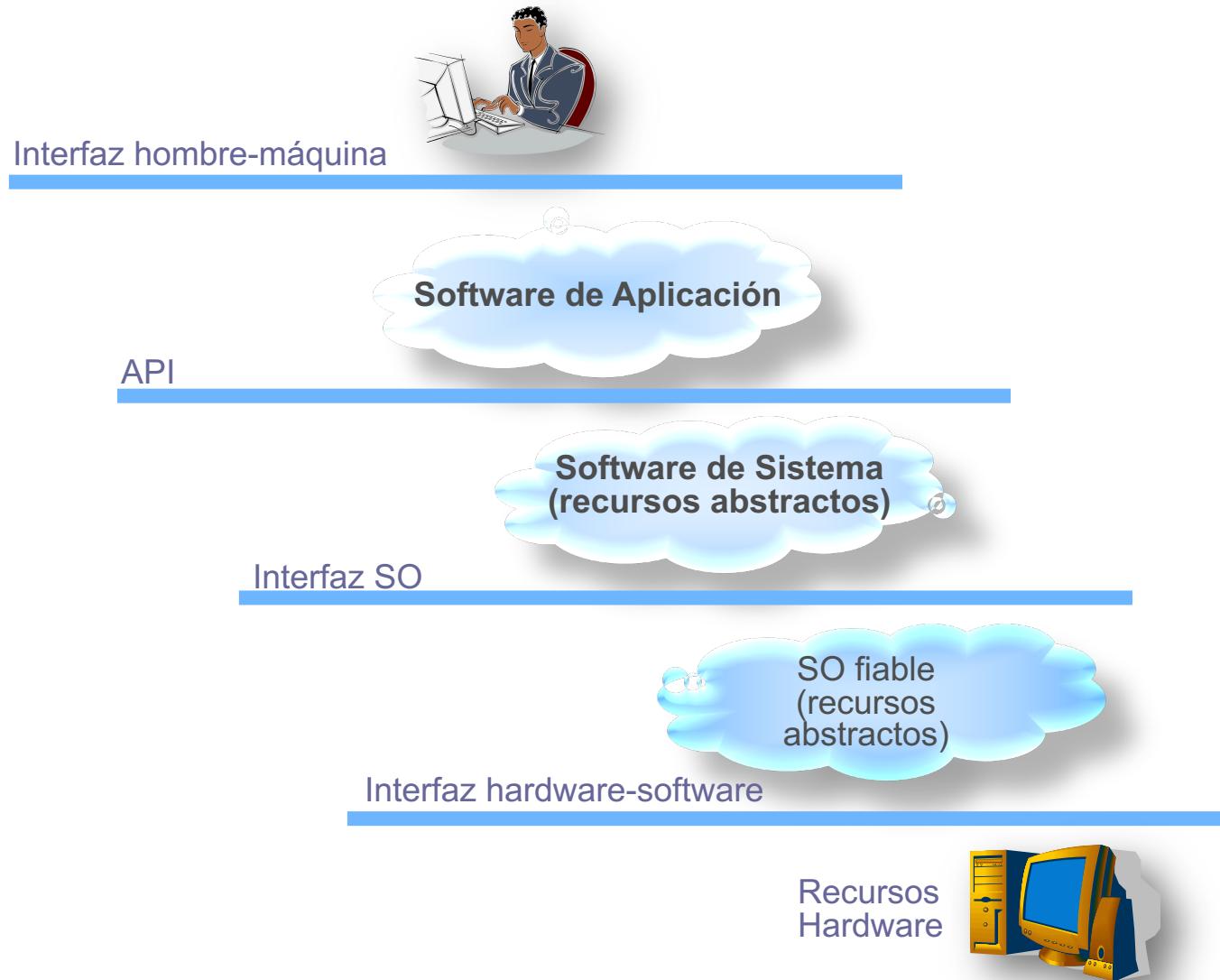
- Según Tanenbaum: lo define según las funciones que cumple
  - Como maquina extendida.
    - Usa hardware y periféricos de una forma sencilla mediante llamadas al Sistema
  - Como administrador de Recursos
    - Administra todos los elementos de un Sistema complejo (Procesadores, memorias, temporizadores, discos, entre otros)
- Según Galvin, Silberschatz:
  - Es aquel programa que se ejecuta continuamente en la computadora (*Kernel*), siendo todo lo demás
    - programas del sistema y
    - programas de aplicación

# El SO como interfaz

- Es una capa entre el usuario y el hardware.
- La interfaz ofrece una máquina extendida que es una **abstracción** de la realidad.



# El SO como interfaz

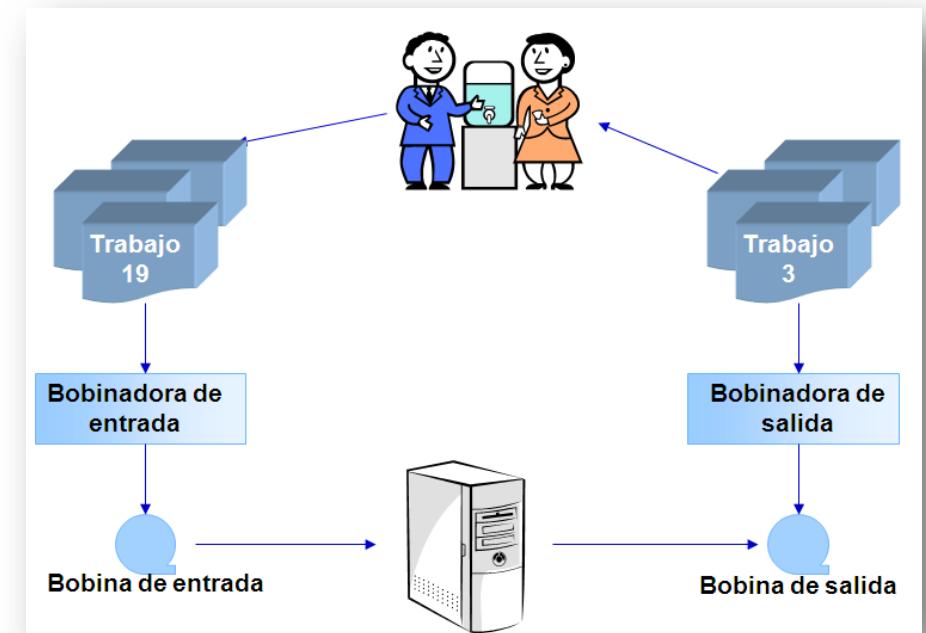


# Estrategias de Sistemas Operativos

- Toda la historia y desarrollo han dejado una amplia variedad de SO de los cuales no todos se conocen en forma exacta
  - **1940.** Primeras computadoras
  - **1950.** Sistemas en Batch
  - **1960.** Multiprogramación y Timesharing
  - **1970.** Minicomputadoras y Microprocesadores.
  - **Finales de 70's y 80's.** Computadoras personales, redes, Sistemas Distribuidos y Sistemas Paralelos, Sistemas de Tiempo Real.
  - **1990. WWW, sistemas de Computo Mobil, PC's.**

# Sistemas de procesamiento por lotes o tandas

- Sirve trabajos individuales de una colección de trabajos predefinidos.
- El SO especifica cada trabajo por medio de una lista de órdenes predefinidas: especificación de control de trabajos
- Usuarios no interactúan con un trabajo.
- Al completar el trabajo, se imprimen los resultados y se devuelve al usuario.



# Sistemas de tiempo compartido

- Varios usuarios interactivamente a la vez
- Usuario establece una sesión interactiva con el computador y proporciona operaciones, programas y datos.
- SO proporciona una respuesta adecuada en tiempo a los usuarios
- Gestión de los recursos y mecanismos de protección
  - Políticas de implementación para compartir equitativamente el procesador
- Sistema multitarea: sistema multiprogramado de tiempo compartido que soporta varios procesos por usuario.

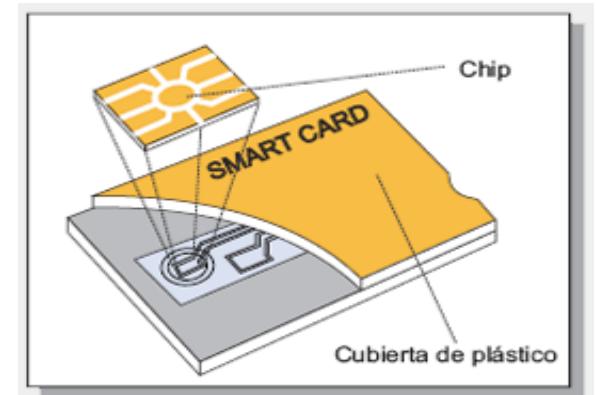
# Computadores personales y estaciones de trabajo

- Máquina completa dedicada a un solo usuario
- Para el SO es más importante minimizar el tiempo de espera para el usuario en vez de maximizar la utilización del hardware.
- El computador puede realizar diferentes tareas concurrentemente.



# Sistemas embebidos

- Concebidos con la idea de controlar sistemas autónomos, carecen de usuario humano
- Funciona como un componente de otro sistema más complejo
- SO debe garantizar tiempos de respuesta para ciertas tareas
- Limitaciones en potencia de procesamiento y memoria.

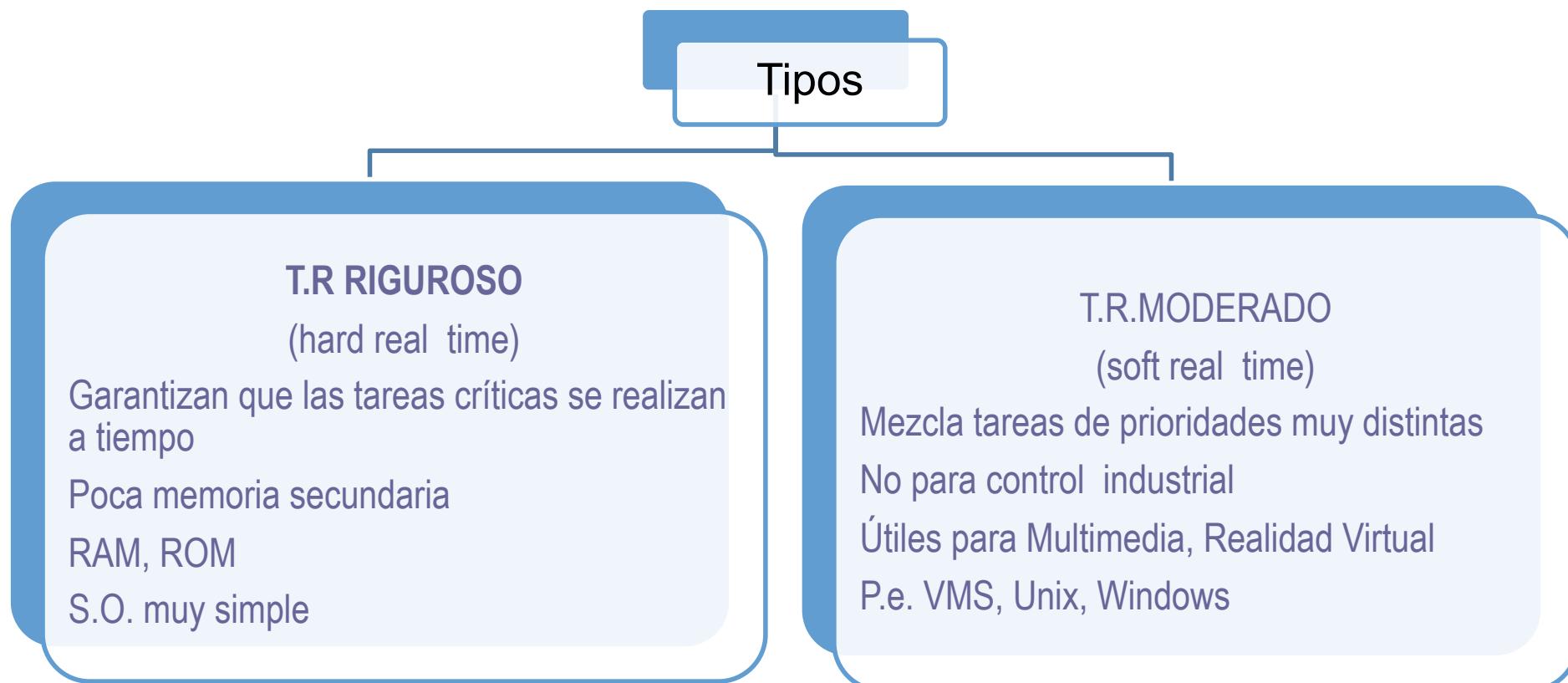


# Sistemas de tiempo real

- Hay sistemas en los que el tiempo de respuesta es crítico
  - Control industrial
  - Control de guiado de misiles
  - Autómatas (air bag, inyección electrónica,...)
  - Asistencia médica

# Sistemas de tiempo real

- “Es aquel en el que si el proceso adecuado no se realiza dentro de unos límites de tiempo muy estrictos y muy pequeños, el sistema falla”.



# Computadores pequeños con capacidades de comunicación (SCC)

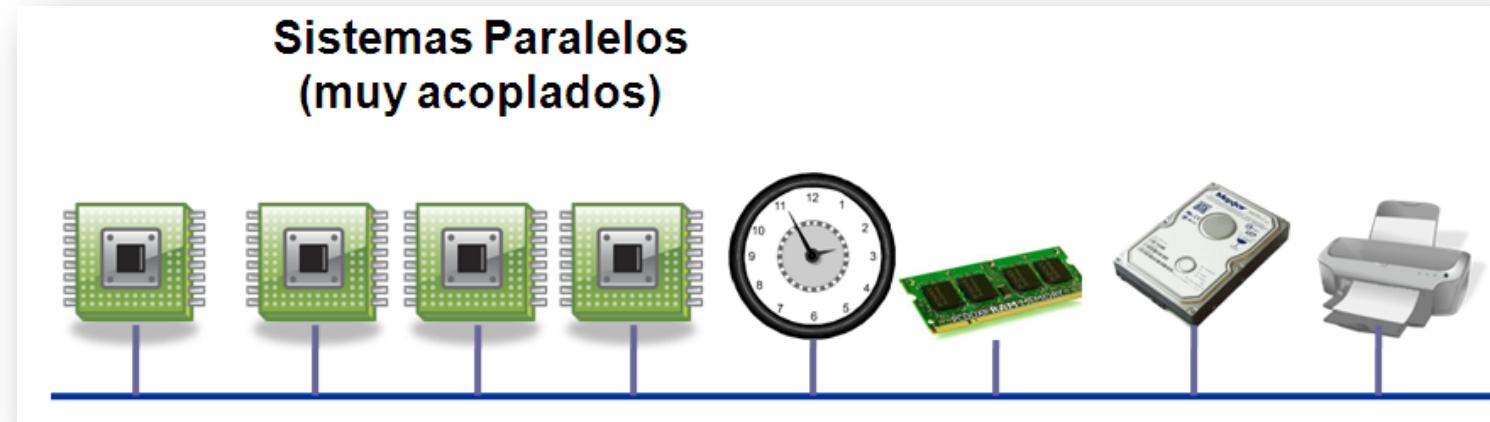
- Incluyen los computadores móviles y los inalámbricos
- Máquinas pequeñas, portátiles, diseñados para las comunicaciones
- SO con nuevas políticas de gestión de recursos, estrategias de gestión de energía, capacidad de almacenamiento limitado, etc.
  - P.e. Tablet, set-top boxes, PDA, smartPhone....
  - PalmOS
  - Windows CE
  - Androi



# Sistemas paralelos

- Para ciertas situaciones se necesita mucha más velocidad

Sistemas  
Tolerantes a Fallos



¡Este sistema no es 4 veces más rápido!

- Comunicación y sincronización
- Conflictos en accesos a recursos comunes
- No todo el código es paralelizable.

# Sistemas distribuidos

- Para cuando se requiere mucha velocidad
  - Distribuir el cálculo
  - + Se comparte recursos
    - + Impresoras, archivos, dispositivos, CPUs,....
  - + Mayor velocidad
  - + Fiabilidad

