

**Lic. en Sistemas de Información**

**SISTEMAS  
OPERATIVOS**

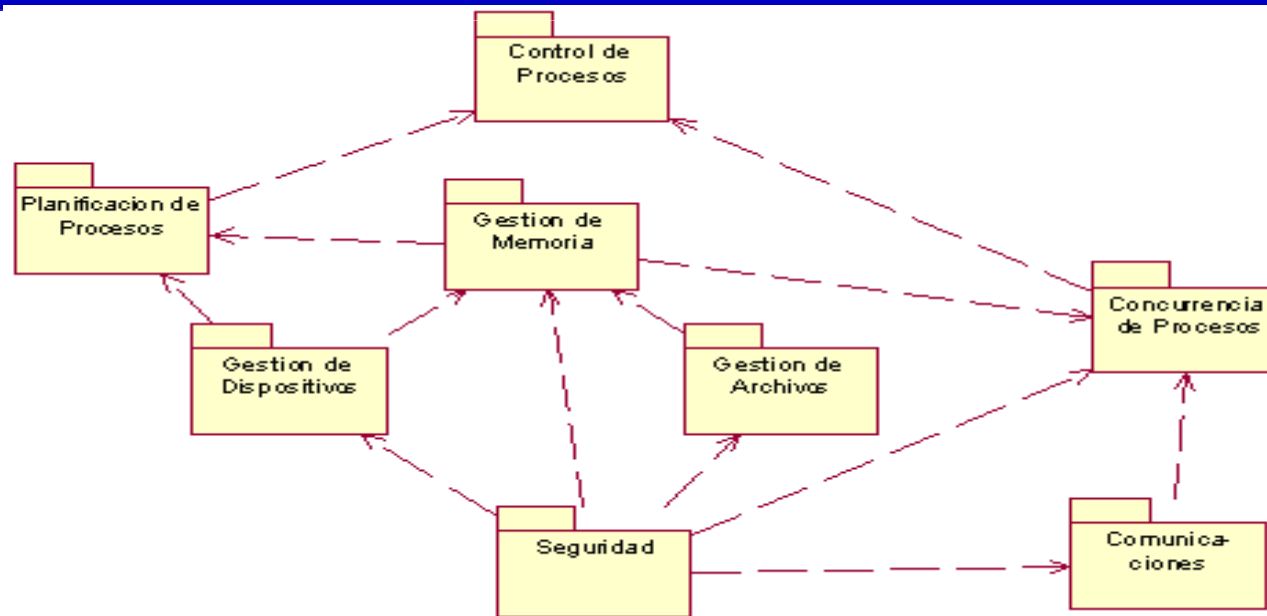
**Ciclo 2014 Plan 2012**

**UNIDAD 1**

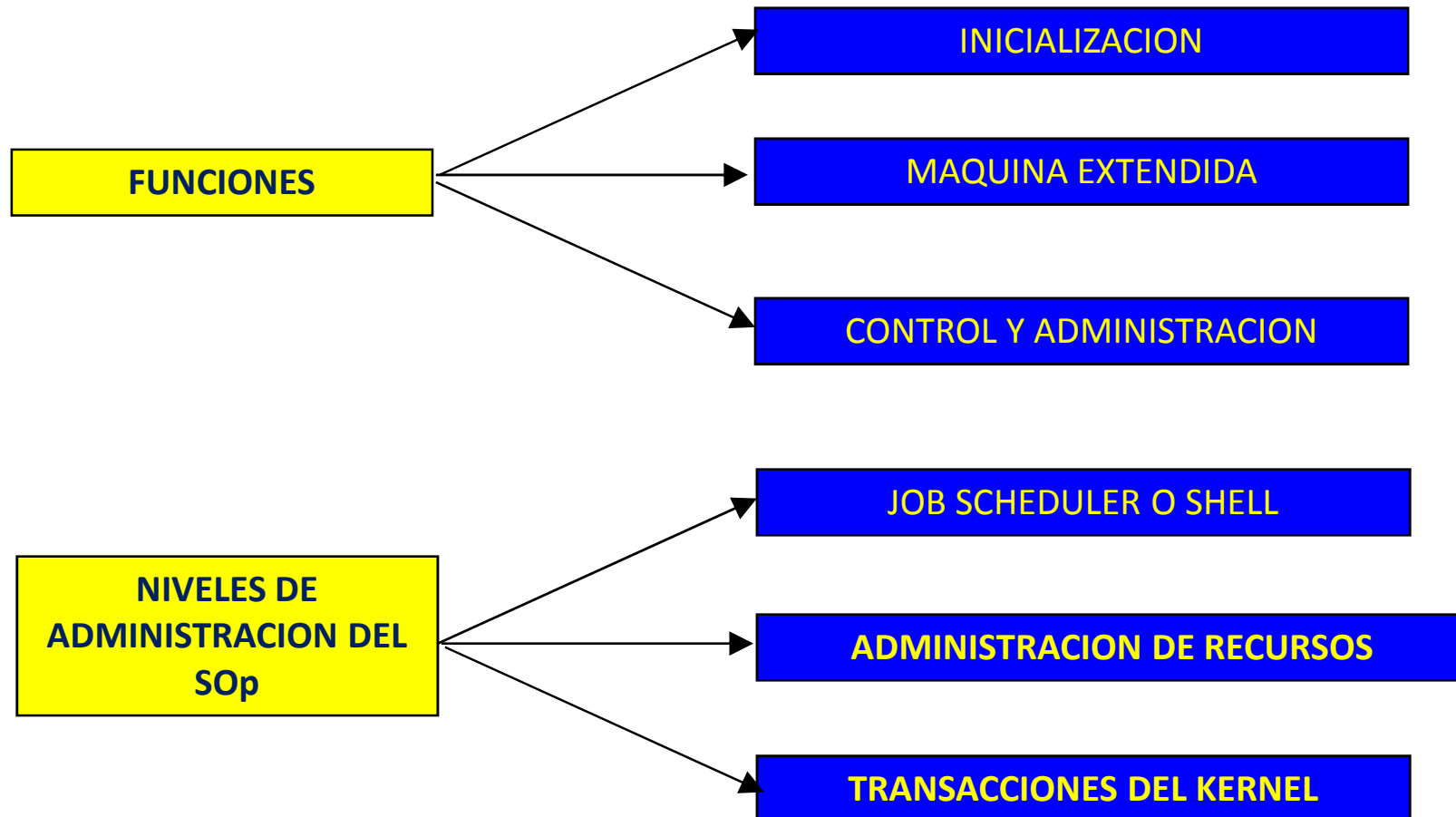
**INTRODUCCION Y CONCEPTOS GENERALES**

## CONCEPTO

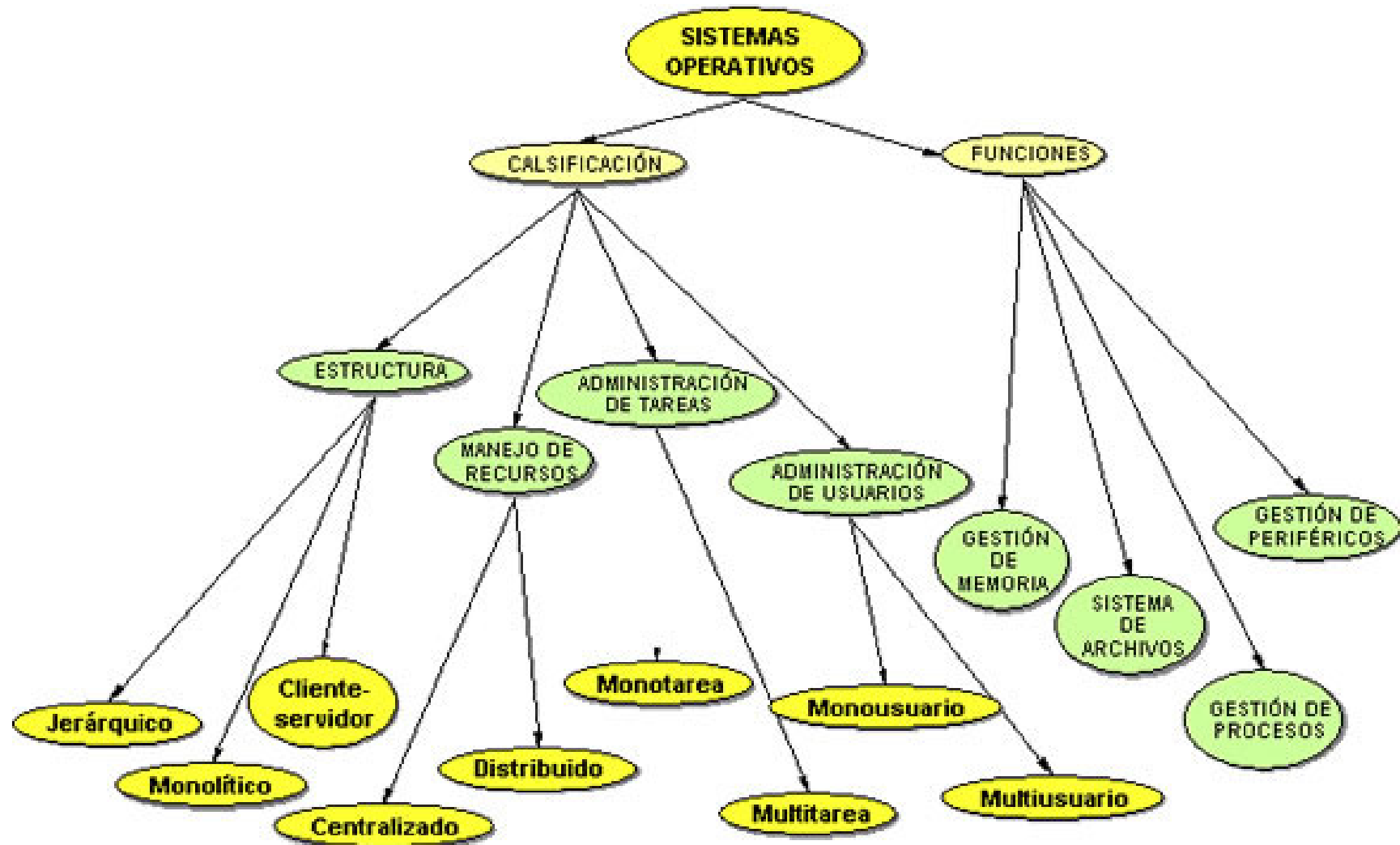
Programa que controla la ejecución de los programas de aplicación, y actúa como interfaz entre las aplicaciones del usuario y el hardware.



**PROPOSITOS DE LOS SISTEMAS OPERATIVOS**



SINOPSIS DE UN SISTEMA OPERATIVO



NIVELES DE PROGRAMACION PARA SOp

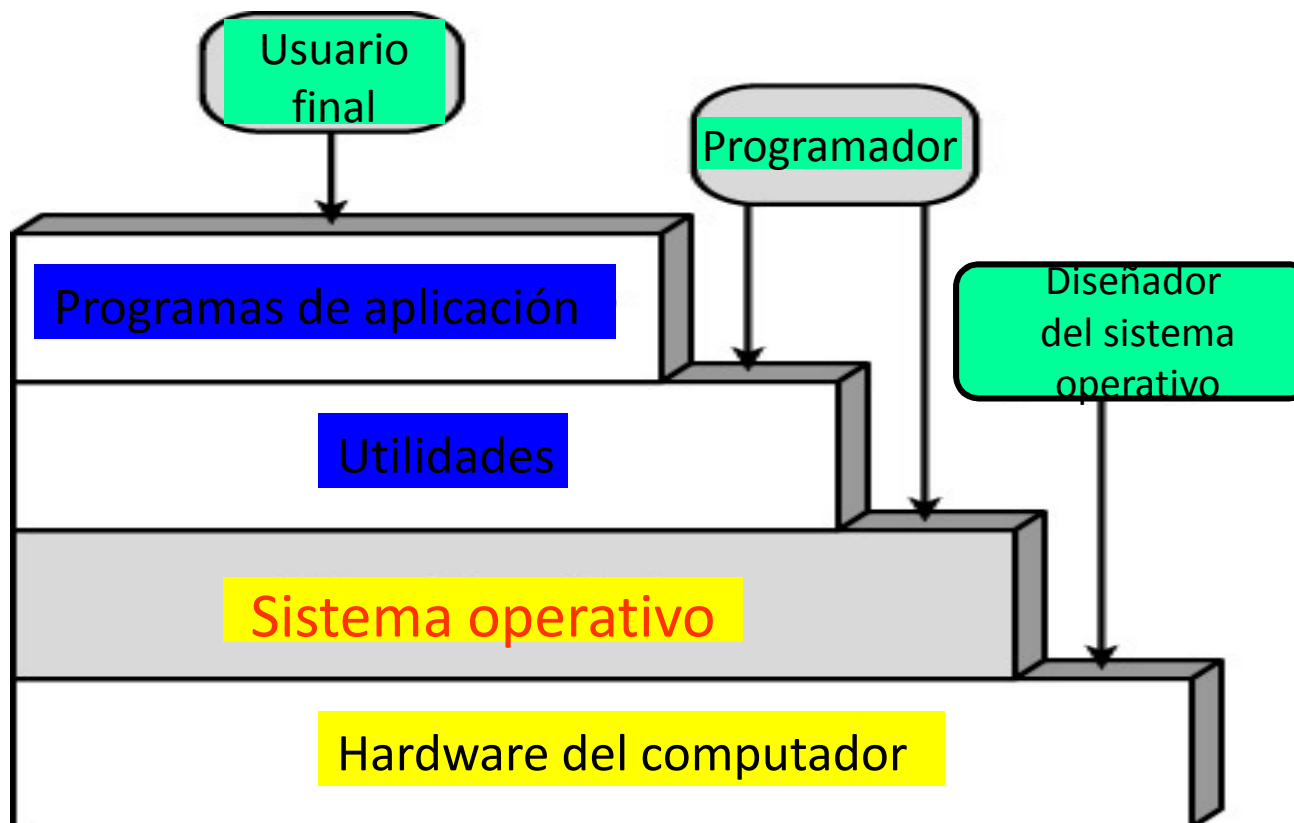
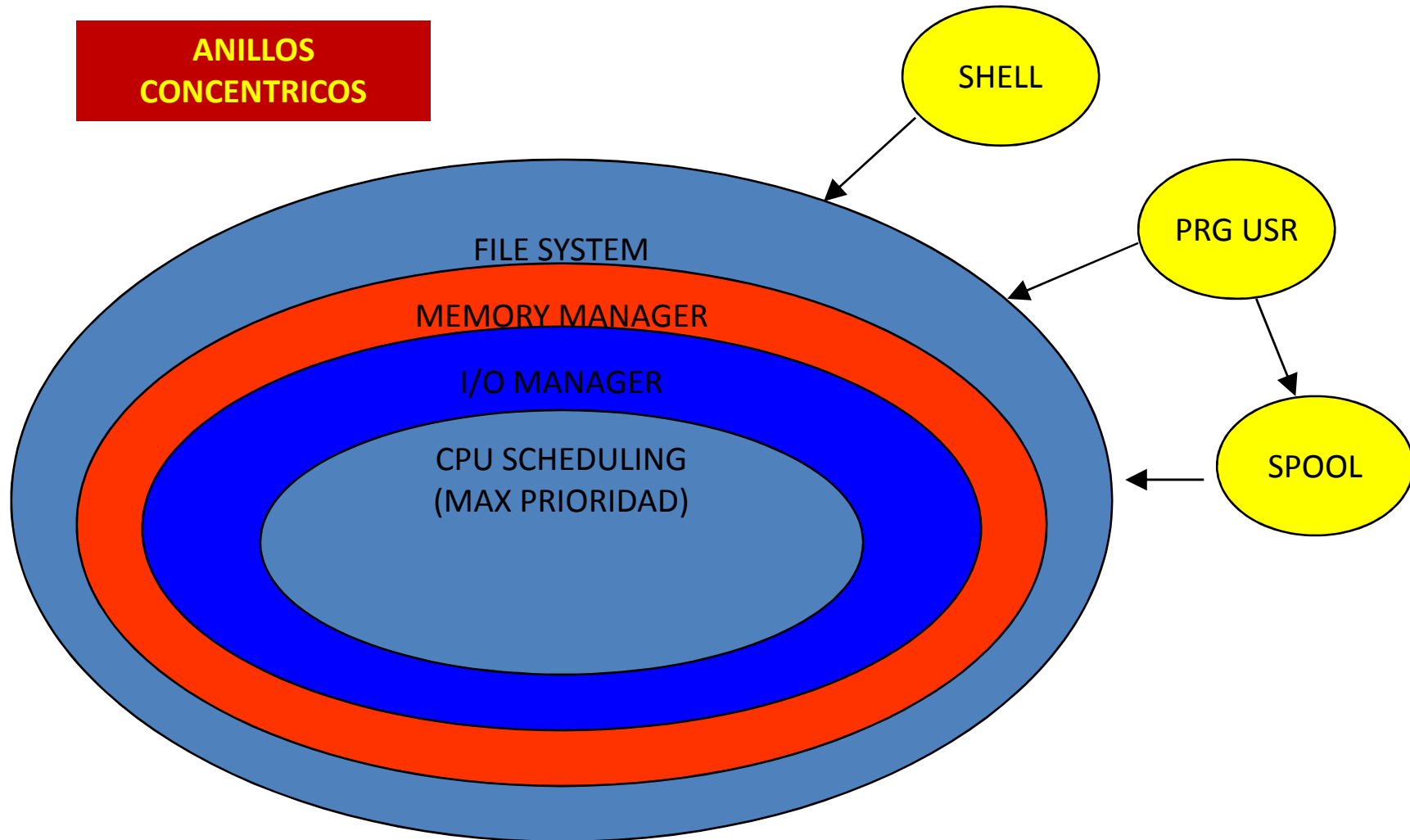


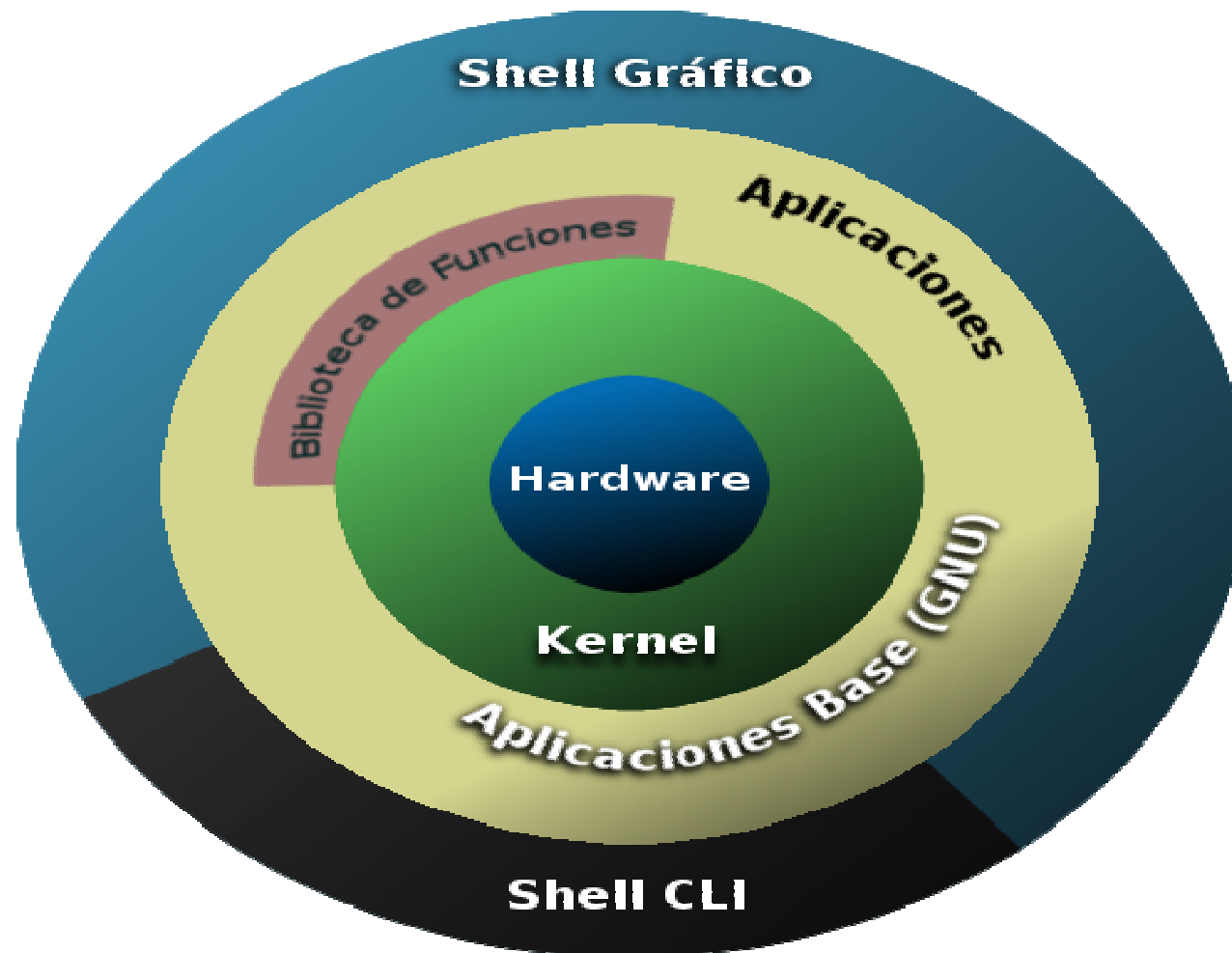
Figura 2.1. Niveles y vistas de un sistema informático.

ARQUITECTURA GENERAL DE LOS SOp

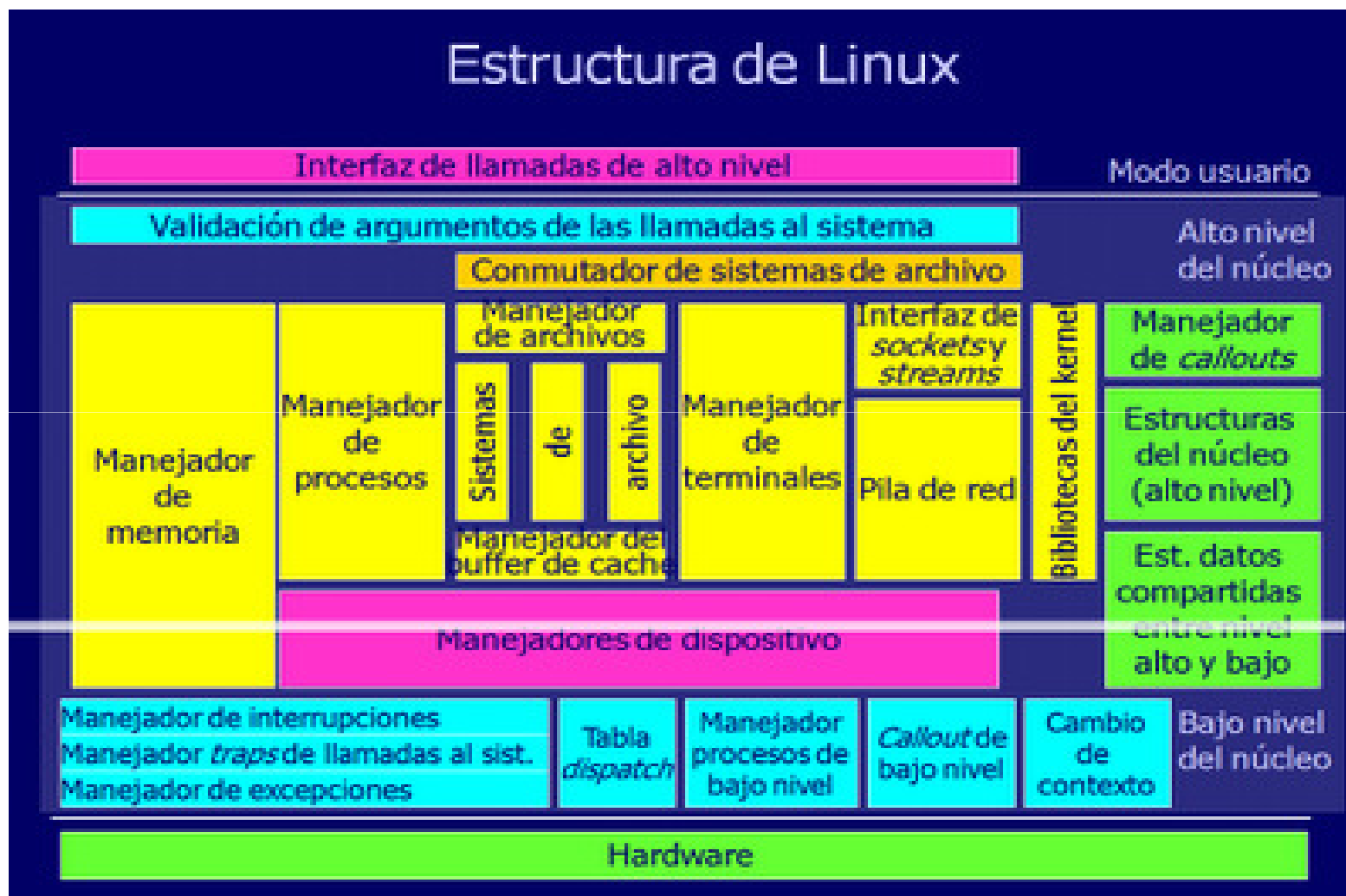
ANILLOS  
CONCENTRICOS



UNIX - LINUX



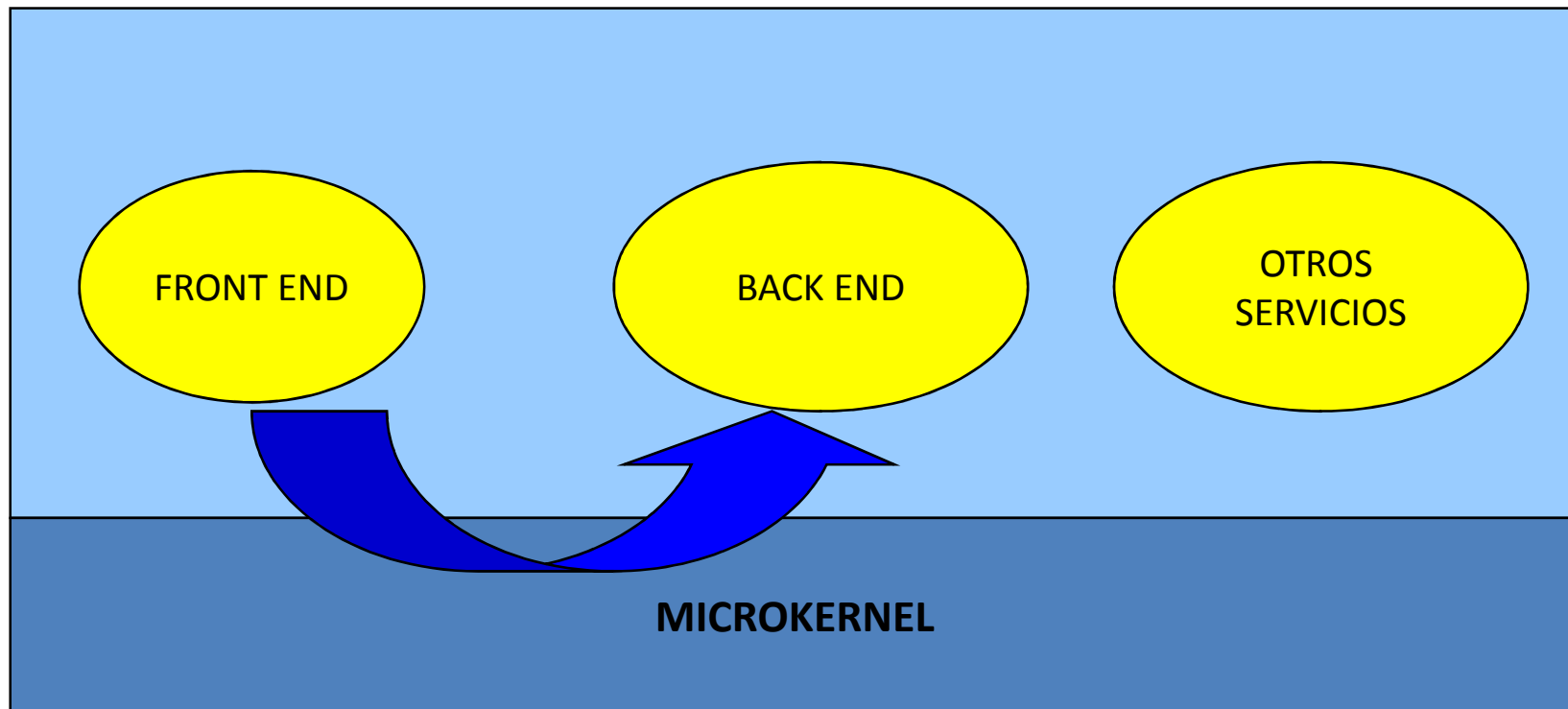
UNIX - LINUX



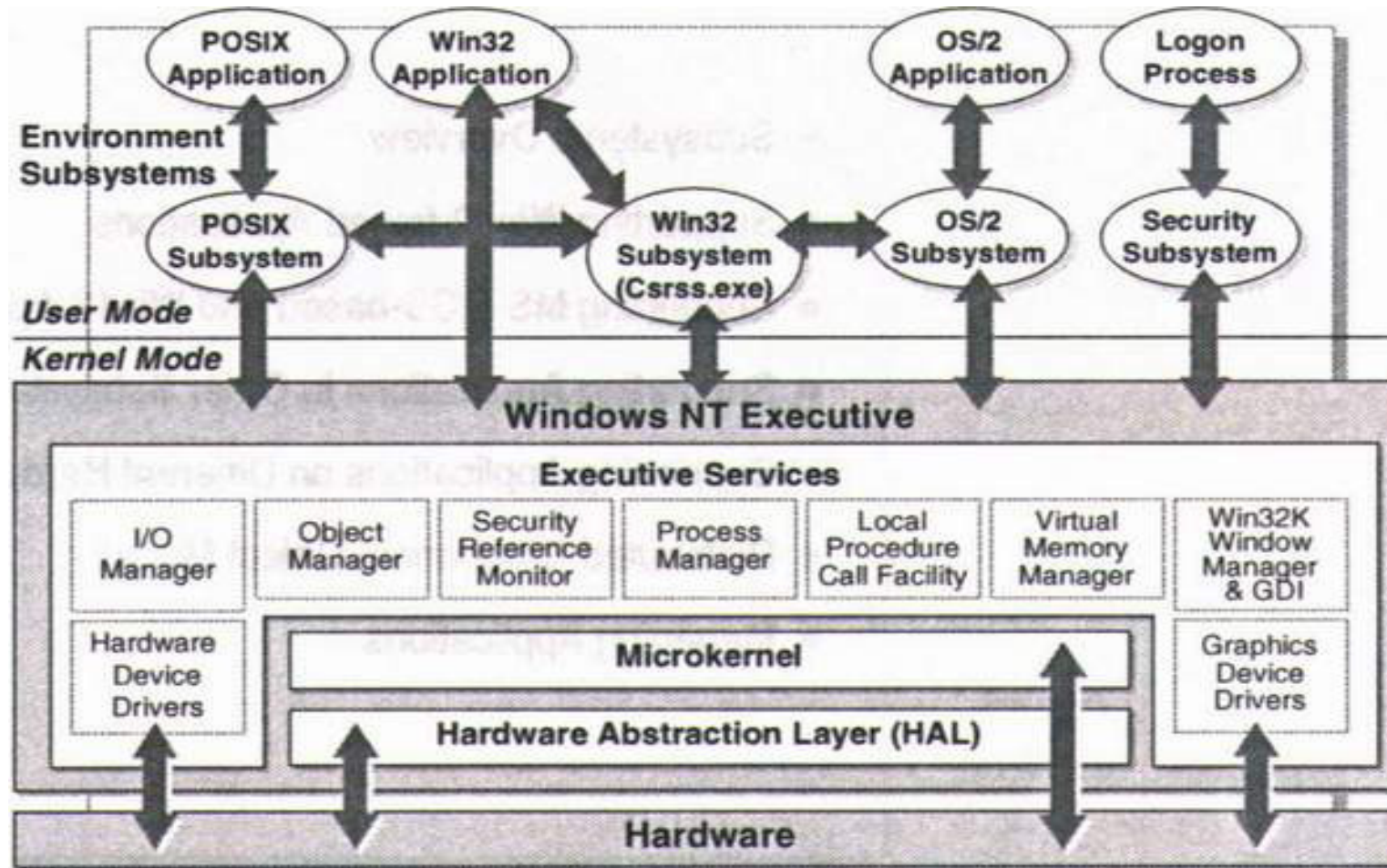


ARQUITECTURA GENERAL DE LOS SOp

CLIENTE/SERVIDOR

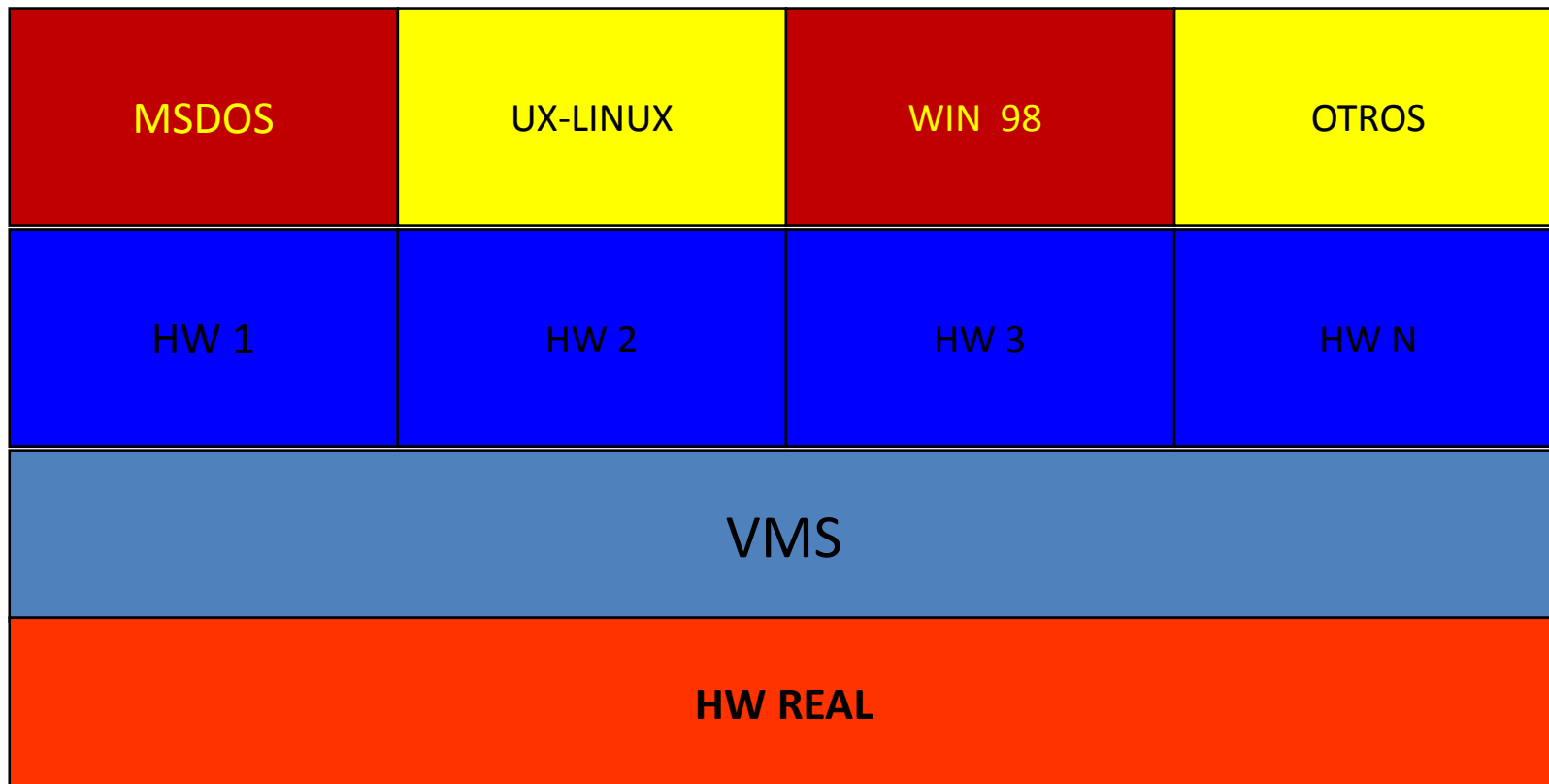


ARQUITECTURA GENERAL DE LOS SOp



ARQUITECTURA GENERAL DE LOS SOp

MAQUINA VIRTUAL



## SISTEMA EMBEBIDO

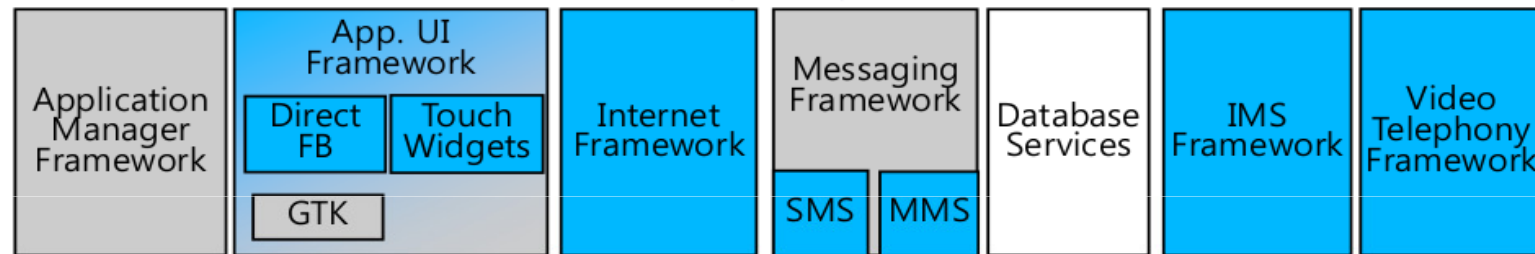
### Reference Applications



Application Manager & UI API

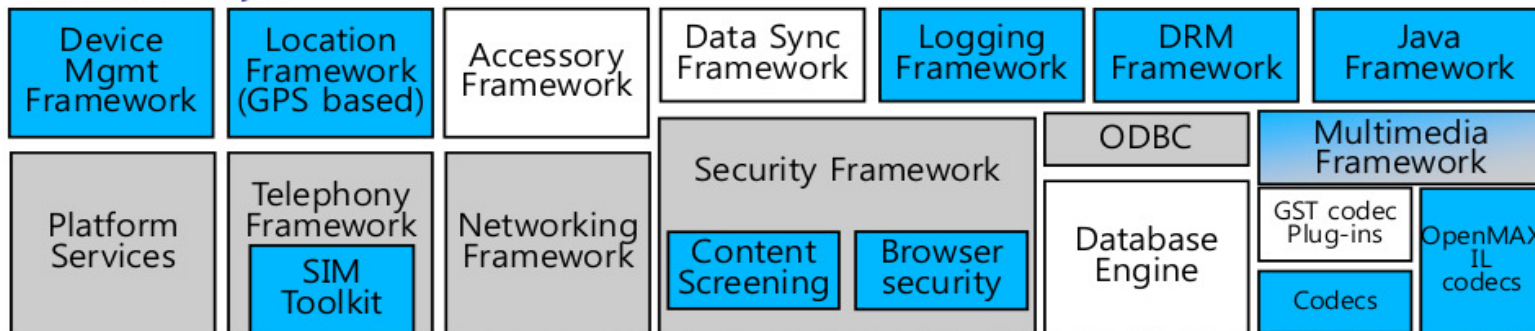
Application Engine API

### Application/UI framework & Application engine Layer



Terminal Service API

### Middleware Layer



### Kernel Layer



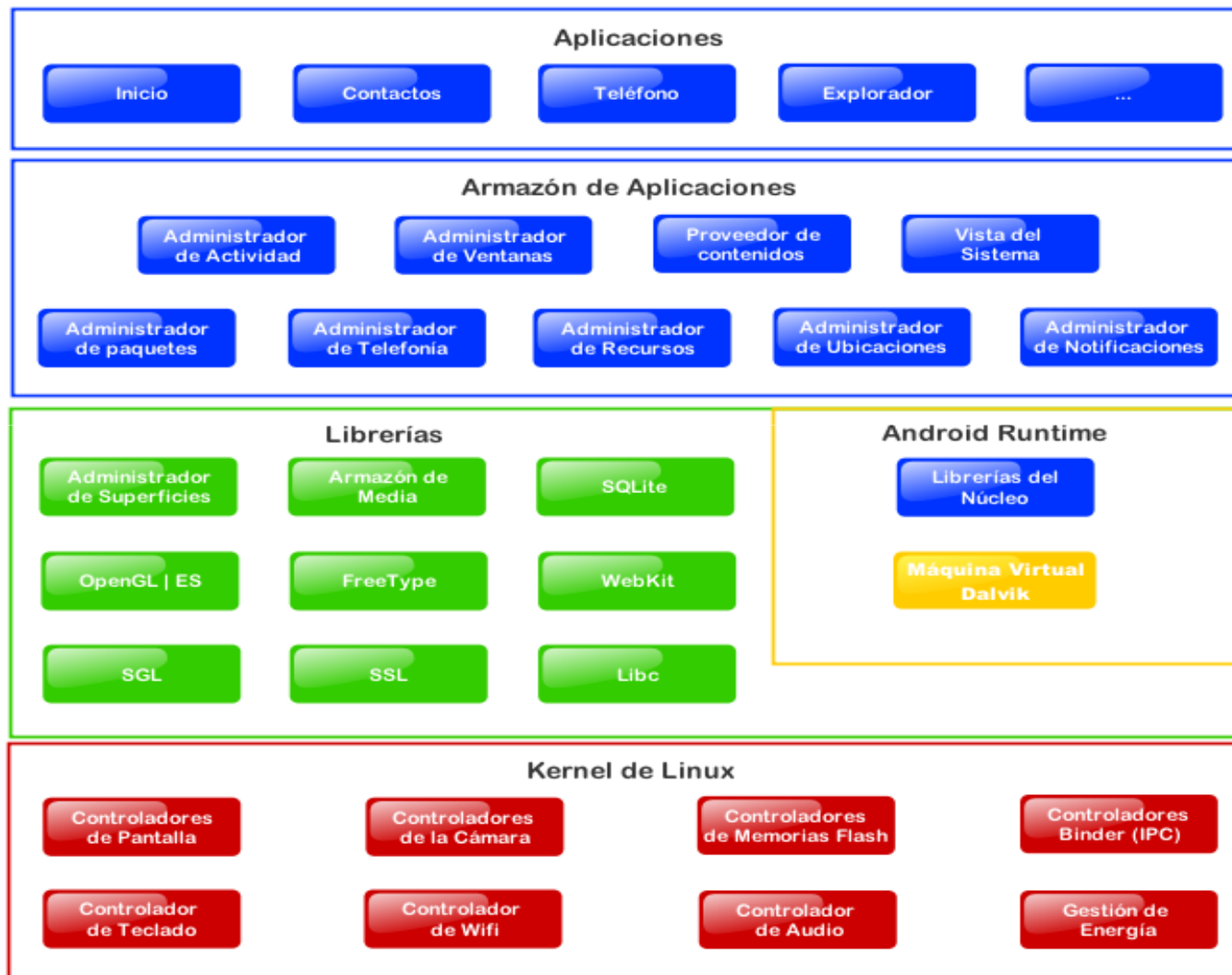
R1 Contribution

R2 Contribution

Available For Contribution

**ARQUITECTURA GENERAL DE LOS SOP**

## ANDROIDE



v  
e

## TIPOS DE SISTEMAS OPERATIVOS

SISTEMA DE PROCESAMIENTO  
POR LOTES

SISTEMA DE TIEMPO  
DIFERIDO

SISTEMA DE TIEMPO REAL

SISTEMA MONOUSUARIO

SISTEMA MULTIUSUARIO

MONOPROCESAMIENTO

MULTIPROCESAMIENTO

CENTRALIZADO

DISTRIBUIDO

PORTABLE

PROPIETARIO

# TIPOS DE SISTEMAS OPERATIVOS

## SISTEMA DE TIEMPO REAL

Quién hace cada módulo:

Microsoft

OEM

OEM independientes



ELEMENTOS PRINCIPALES DEL SISTEMA OPERATIVO

JERARQUIA DE DISEÑO

Nivel	Nombre	Objetos	Ejemplos de operaciones
13	Shell	Entorno de programación de usuario	Sentencias de un lenguaje de shell
12	Procesos de usuario	Procesos de usuario	Salir, eliminar, suspender, reanudar
11	Directorios	Directorios	Crear, destruir, conectar, desconectar, buscar, listar
10	Dispositivos	Dispositivos externos tales como impresoras, pantallas y teclados	Abrir, cerrar, leer, escribir
9	Sistema de archivos	Archivos	Crear, destruir, abrir, cerrar, leer, escribir
8	Comunicaciones	Tubos ( <i>pipes</i> )	Crear, destruir, abrir, cerrar, leer, escribir



ELEMENTOS PRINCIPALES DEL SISTEMA OPERATIVO

JERARQUIA DE DISEÑO

Nivel	Nombre	Objetos	Ejemplos de operaciones
7	Memoria virtual	Segmentos, páginas	Leer, escribir, traer ( <i>fetch</i> )
6	Almacenamiento secundario local	Bloques de datos, canales de dispositivos	Leer, escribir, asignar, liberar
5	Procesos primitivos	Procesos primitivos,	Suspender, reanudar, esperar, semáforos, colas de señalizar procesos listos

ELEMENTOS PRINCIPALES DEL SISTEMA OPERATIVO

JERARQUIA DE DISEÑO

Nivel	Nombre	Objetos	Ejemplos de operaciones
4	Interrupciones	Programas de tratamiento de interrupciones	Invocar, enmascarar, desenmascarar, reintentar
3	Procedimientos llamadas, visualización	Procedimientos, pila de retornar	Marcar la pila, llamar,
2	Conjunto de instrucciones	Evaluación de la pila, intérprete de microprogramas, vectores de datos y escalares	Cargar, almacenar, sumar, restar, bifurcar
1	Circuitos electrónicos	Registros, puertas, buses, etc.	Borrar, transferir, activar, complementar

## CARACTERISTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS:

- Arquitectura micronúcleo:

- Asigna solamente unas pocas funciones esenciales al núcleo.

- Espacios de direcciones.
- Comunicación entre procesos (*IPC*).
- Planificación básica.

- Multihilos:

- El proceso se divide en hilos que pueden ejecutarse concurrentemente.

- Hilo:

- Unidad de trabajo que se puede expedir para su ejecución.
- Se ejecuta secuencialmente y es interrumpible.

- Proceso:

- Un conjunto de uno o más hilos.

## CARACTERISTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS:

- Multiproceso simétrico (*SMP*):
  - Existencia de múltiples procesadores.
  - Estos procesadores comparten la misma memoria principal y dispositivos de E/S.
  - Todos los procesadores pueden ejecutar las mismas funciones.
  
- Sistema operativo distribuido:
  - Proporciona la ilusión de un único espacio de memoria principal y un único espacio de memoria secundaria.
  - Utilizado para el sistema de archivos distribuido.

## CARACTERISTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS:

- Diseño orientado a objetos:
  - Añade extensiones modulares a un pequeño núcleo.
  - Permite a los programadores personalizar un sistema operativo sin romper la integridad del sistema.

**CARACTERISTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS:**

**GESTION DE MEMORIA**

- Aislamiento del proceso.
- Asignación y gestión automáticas.
- Soporte para la programación modular.
  - Protección y control de acceso.
  - Almacenamiento a largo plazo.

**MEMORIA VIRTUAL**

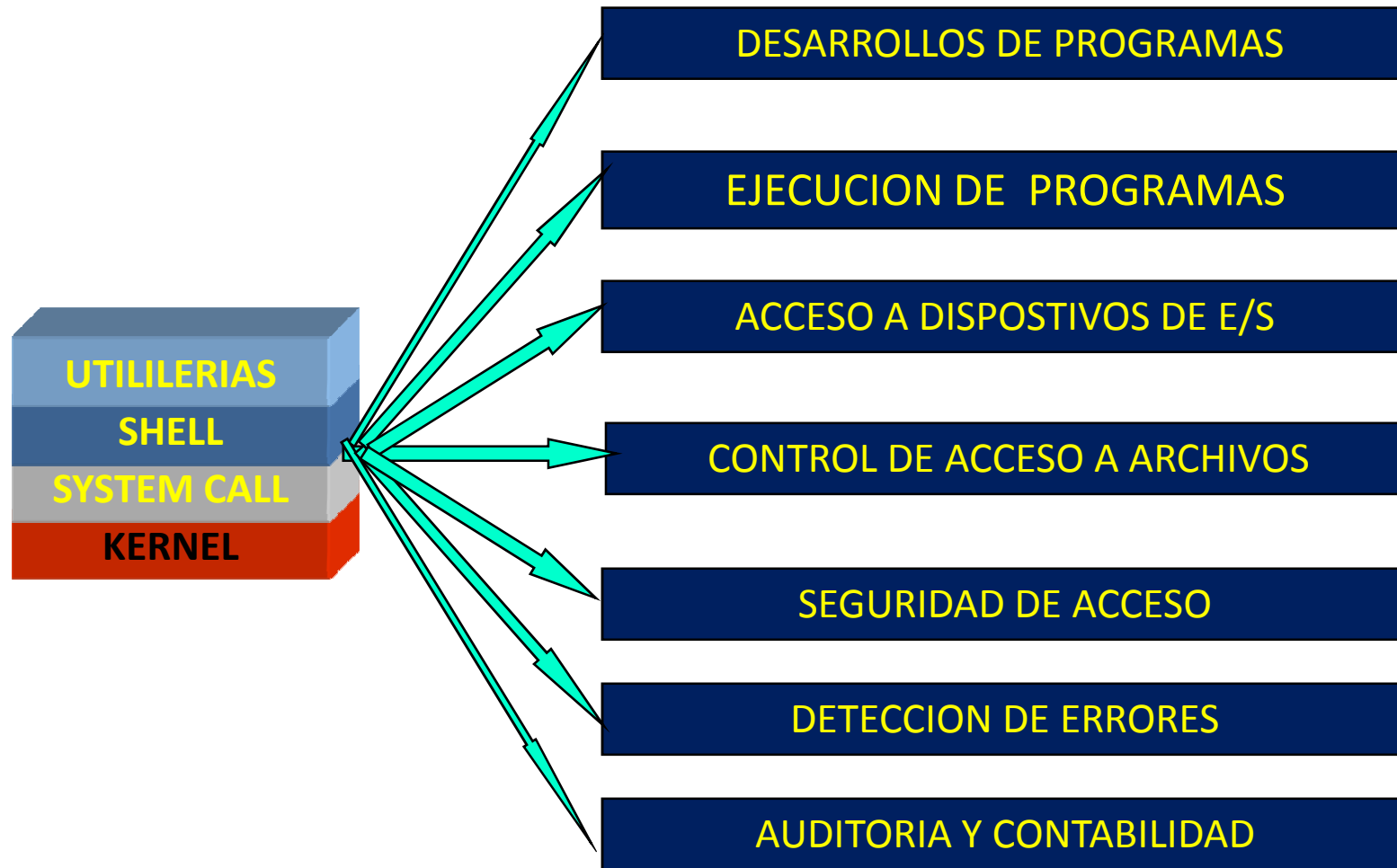
- Permite a los programas direccionar la memoria desde un punto de vista lógico.
- No existirá un espacio muerto entre la ejecución de los procesos sucesivos, mientras un proceso se envía al almacenamiento secundario y el proceso que le sucede es traído de éste.

CARACTERISTICAS DE LOS SISTEMAS OPERATIVOS MODERNOS:

SEGURIDAD /PROTECCION

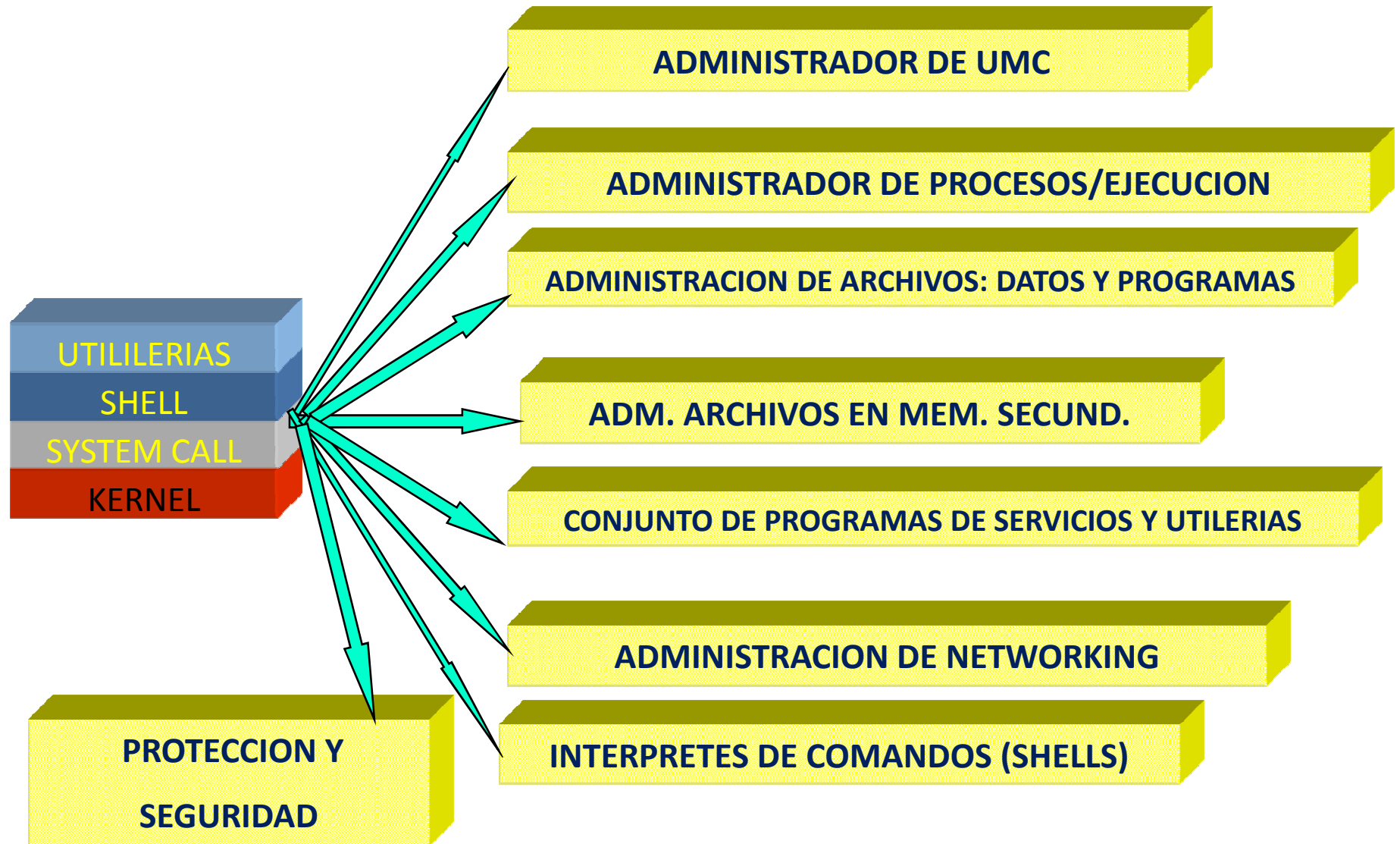
- Control de acceso:
  - Regula el acceso del usuario al sistema.
- Control del flujo de información:
  - Regula el flujo de datos dentro del sistema y su distribución a los usuarios.
- Certificación:
  - Demuestra que el acceso y el control del flujo se llevan a cabo de acuerdo con las especificaciones.

SERVICIOS DE UN SOp



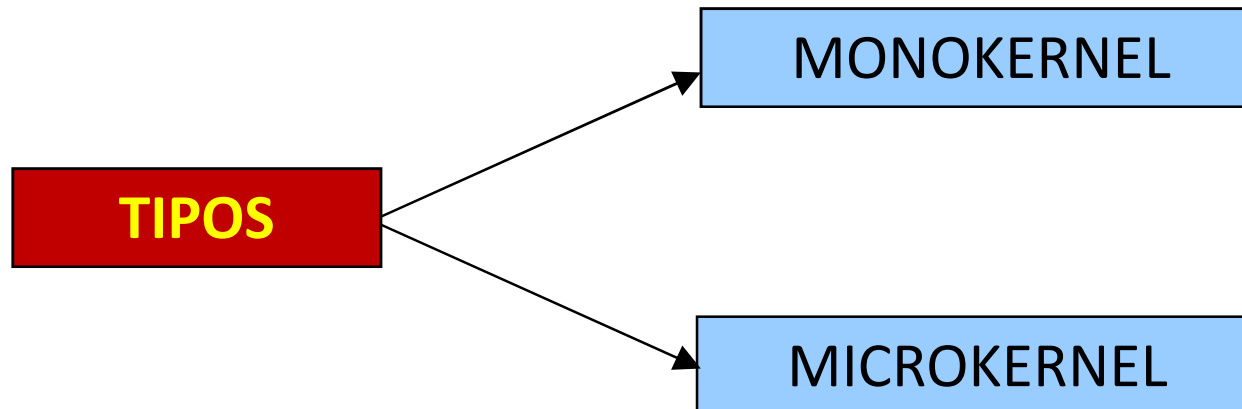


**ADMINISTRADORES DE UN SOp**



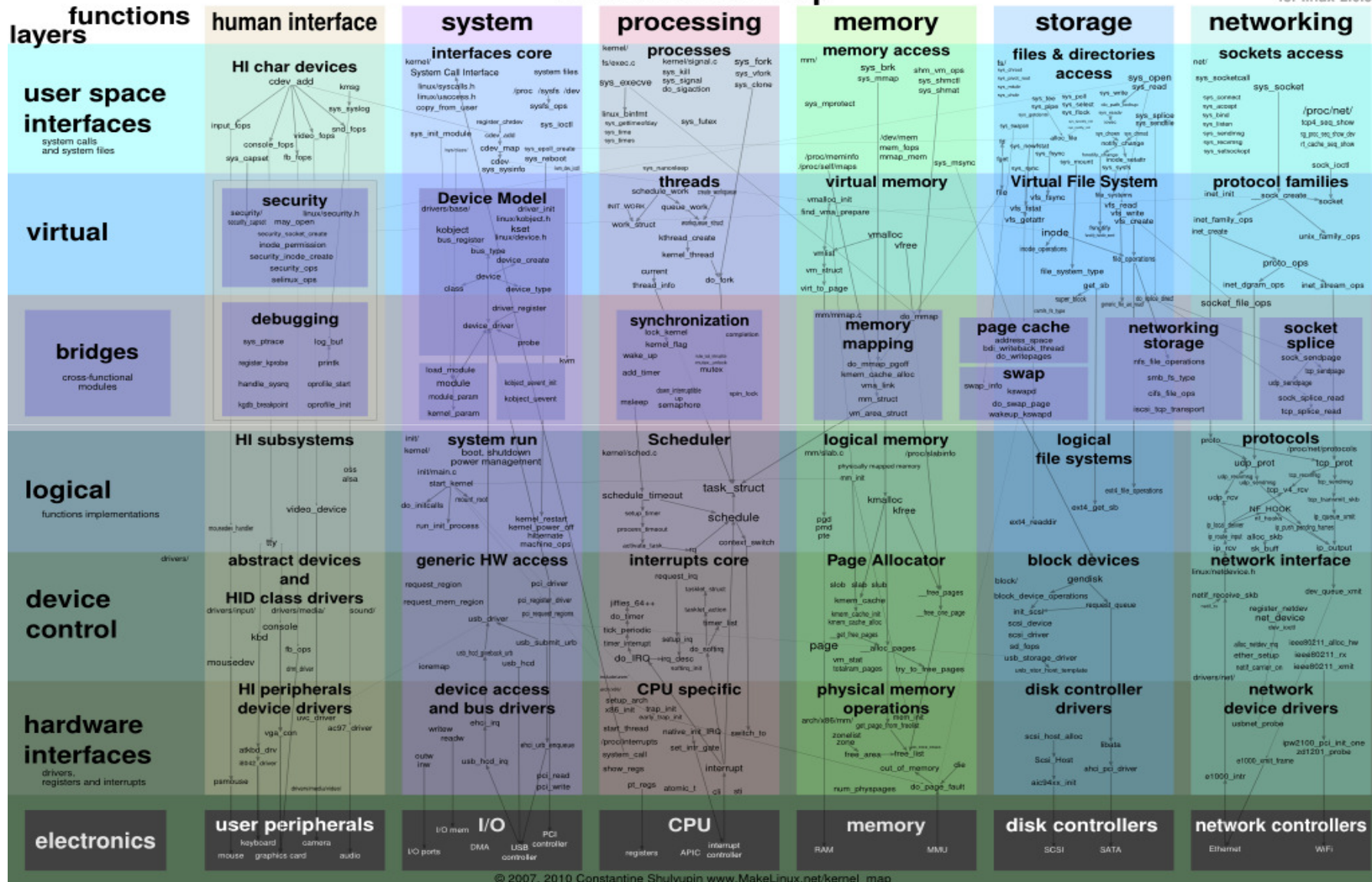
## NUCLEO (KERNEL)

- Parte del sistema operativo que se encuentra en la memoria principal.
- Incluye las funciones utilizadas con más frecuencia.
- También denominado *kernel*.



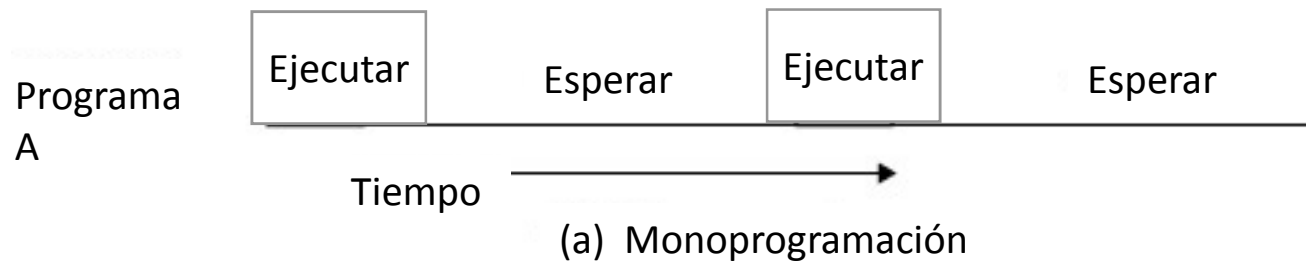
## Linux kernel map

for linux-2.6.36



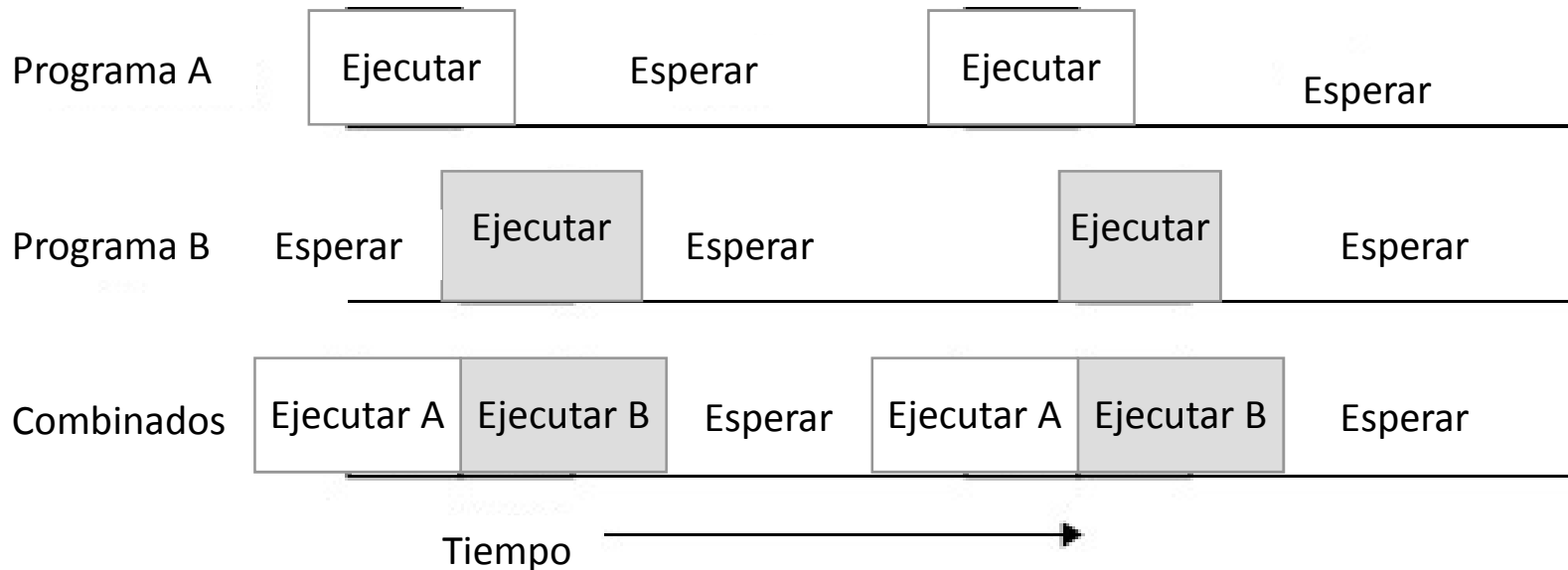
## TERMINOLOGIA BASICA: MONOPROGRAMACION

Antes de continuar, el procesador debe esperar hasta que la instrucción de E/S termine.



## TERMINOLOGIA BASICA: MULTIPROGRAMACION

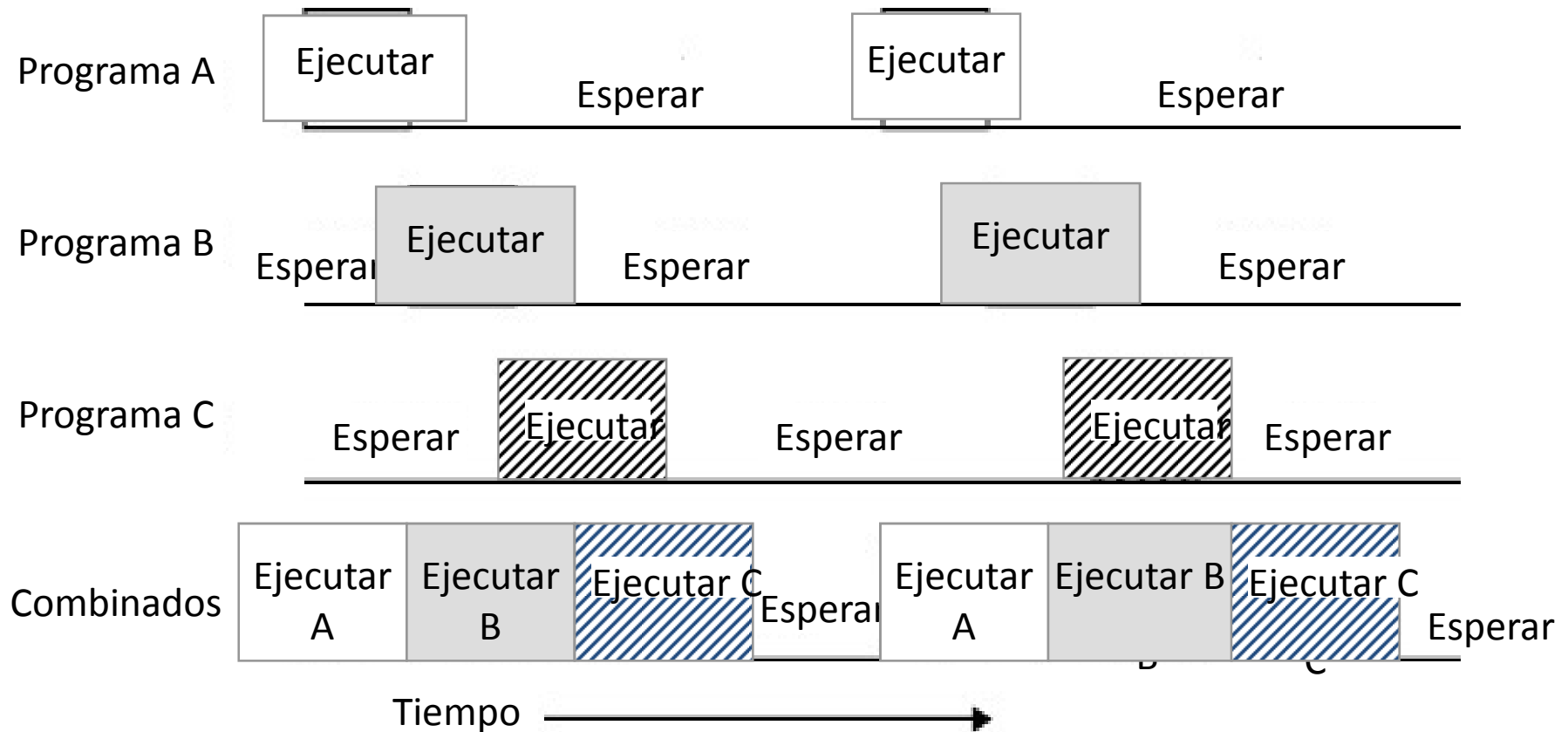
- Cuando un trabajo necesite esperar una E/S, el procesador puede cambiar al otro trabajo.



Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4

(b) Multiprogramación con dos programas

**TERMINOLOGIA BASICA:  
MULTIPROGRAMACION**



(c) Multiprogramación con tres programas

TERMINOLOGIA BASICA: MULTIPROGRAMACION

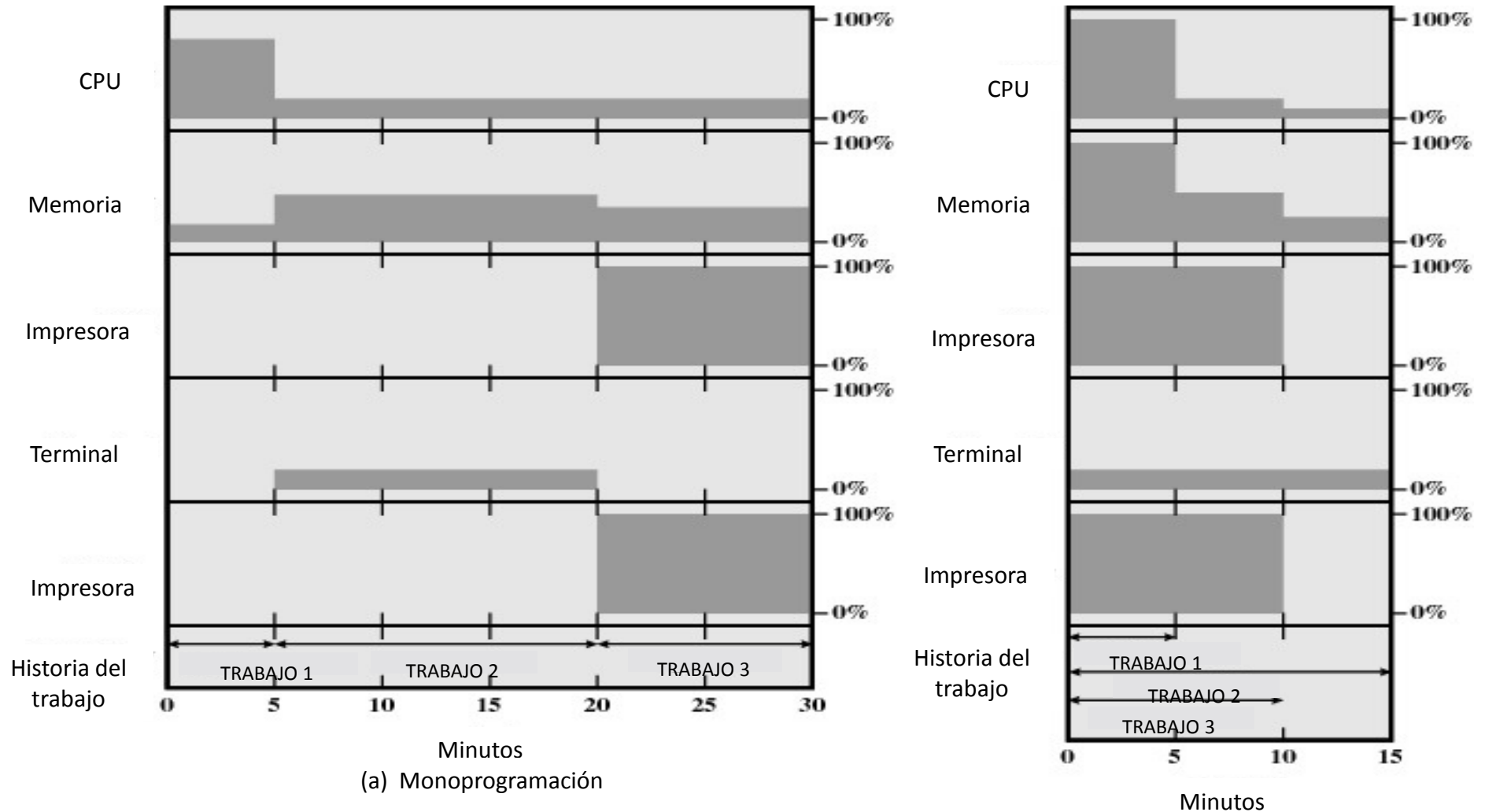


Figura 2.6. Histograma de utilización.

Williams Stallings SISTEMAS OPERATIVOS. Principios de diseño e interioridades. 4ta ed. Pearson Educación S.A. Madrid, 2001 ISBN: 84-205-3177-4



MULTIPROGRAMACION: EJEMPLO Y RESULTADOS

	TRABAJO1	TRABAJO2	TRABAJO3
Tipo de trabajo	Cálculo intensivo	E/S intensiva	E/S intensiva
Duración	5 min.	15 min.	10 min.
Memoria exigida	50 K	100 K	80 K
¿Necesita disco?	No	No	Sí
¿Necesita terminal?	No	Sí	No
¿Necesita impresora?	No	No	Sí

	Monoprogramación	Multiprogramación
Uso del procesador	22%	43%
Uso de la memoria	30%	67%
Uso del disco	33%	67%
Uso de la impresora	33%	67%
Tiempo transcurrido	30 min.	15 min.
Tasa de productividad	6 trabajos/hora	12 trabajos/hora
Tiempo medio de respuesta	18 min.	10 min.



TERMINOLOGIA BASICA: MULTIPROGRAMACION

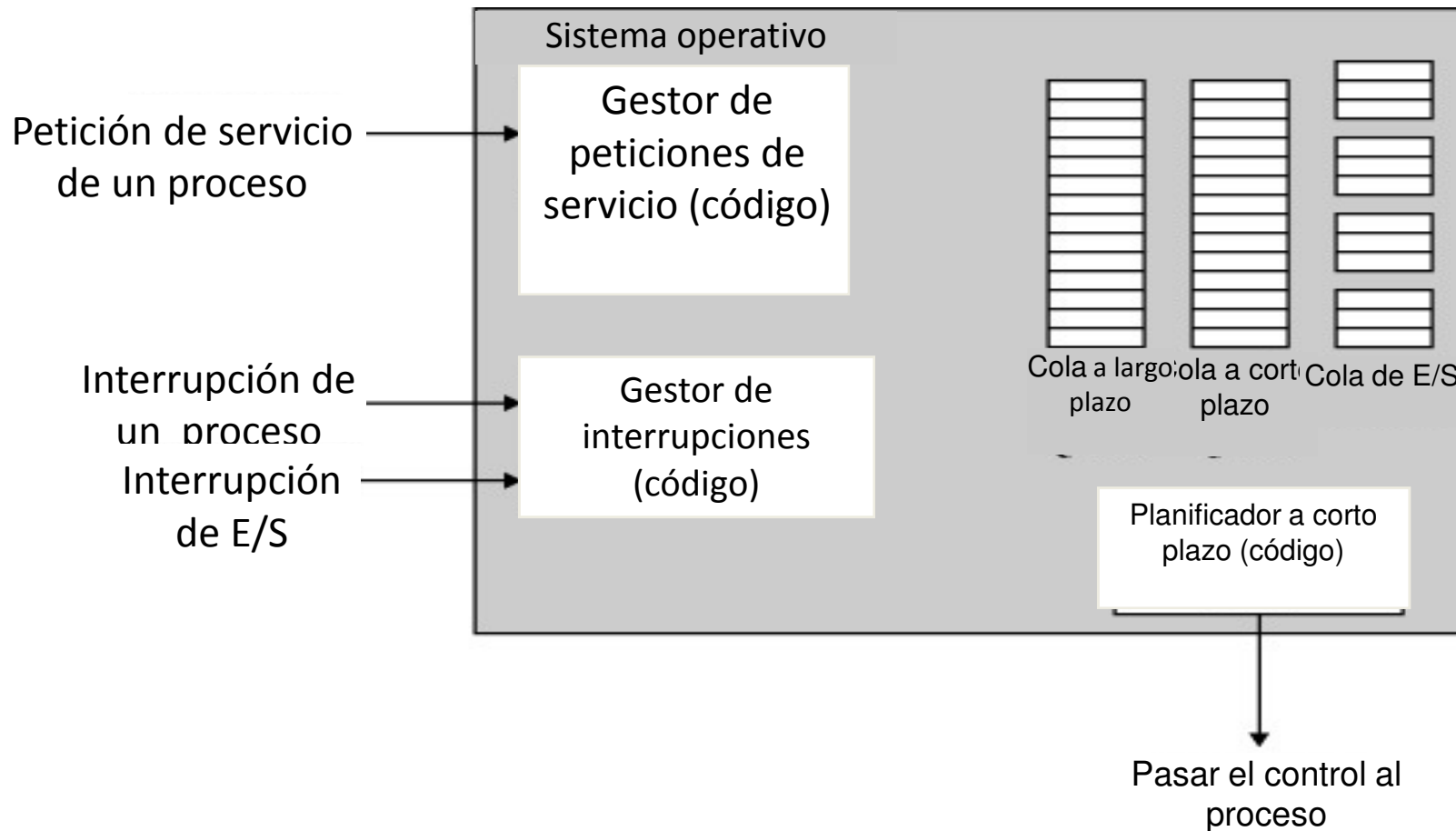


Figura 2.11. Elementos clave para la multiprogramación de un sistema operativo.

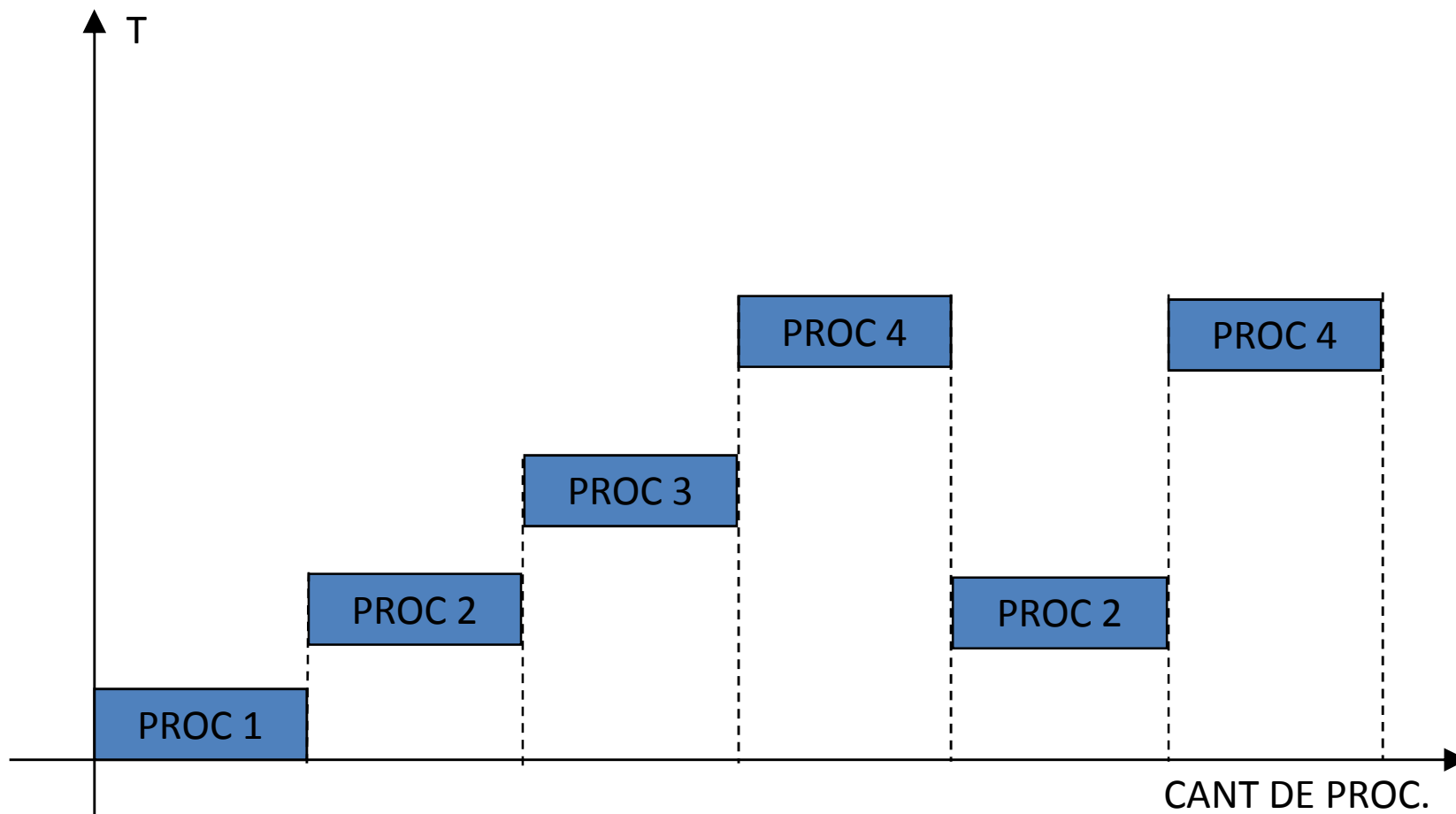
**TERMINOLOGIA BASICA: MULTITAREA**

- Utiliza la multiprogramación para gestionar varias tareas interactivas.
- El tiempo del procesador se comparte entre los diversos usuarios.
- Múltiples usuarios acceden simultáneamente al sistema por medio de terminales.

	<b>Multiprogramación por lotes</b>	<b>Tiempo compartido</b>
<b>Objetivo principal</b>	<b>Maximizar la utilización del procesador</b>	<b>Minimizar tiempo de respuesta</b>
<b>Origen de las instrucciones al sistema operativo</b>	<b>Instrucciones de un lenguaje de control de trabajos incluidas en el trabajo</b>	<b>Órdenes dadas en el terminal</b>

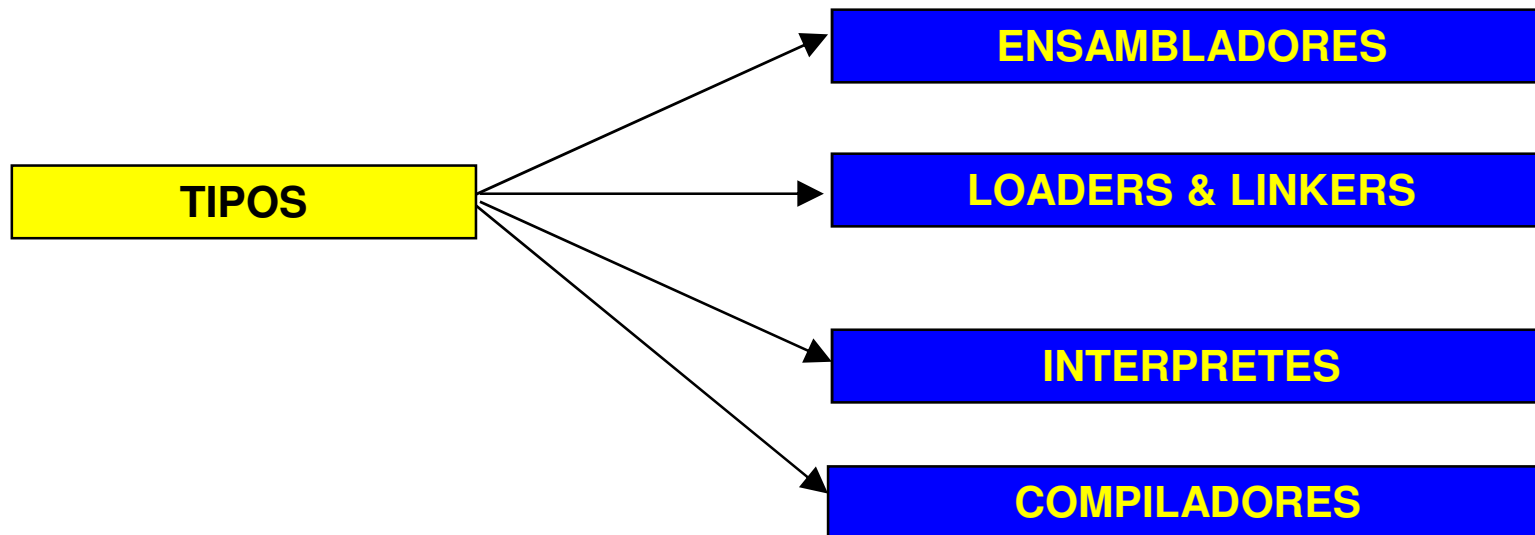
TERMINOLOGIA BASICA: MULTITAREA

EJECUCION BAJO ALGORITMOS DE TIEMPO COMPARTIDO



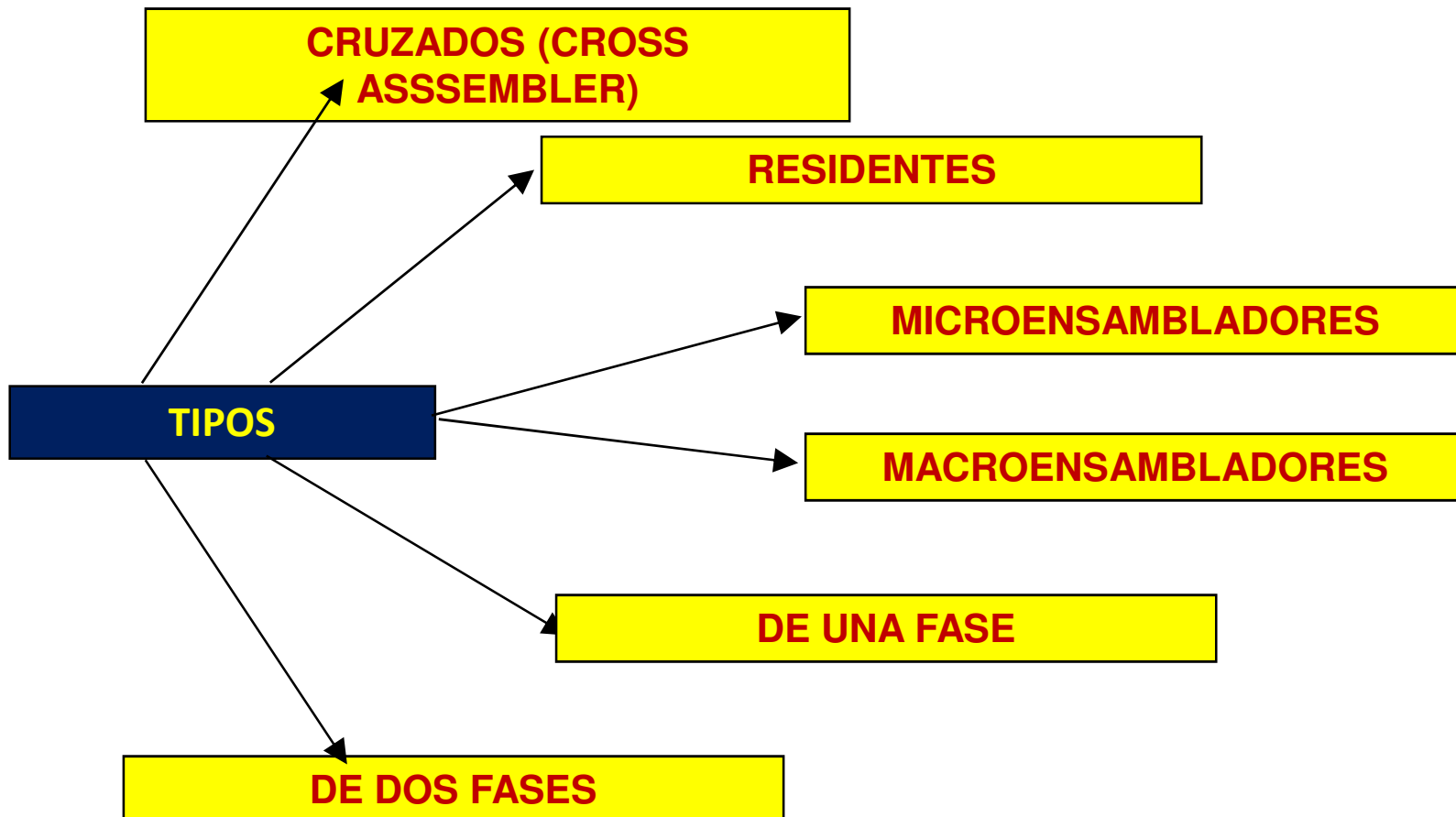
**SOFTWARE DE SISTEMA**

**PROGRAMAS QUE CONFORMAN UTILERIAS DEL SISTEMA OPERATIVO,  
QUE LE PERMITEN REALIZAR TAREAS INTERNAS DE OPERACION**



**ENSAMBLADORES**

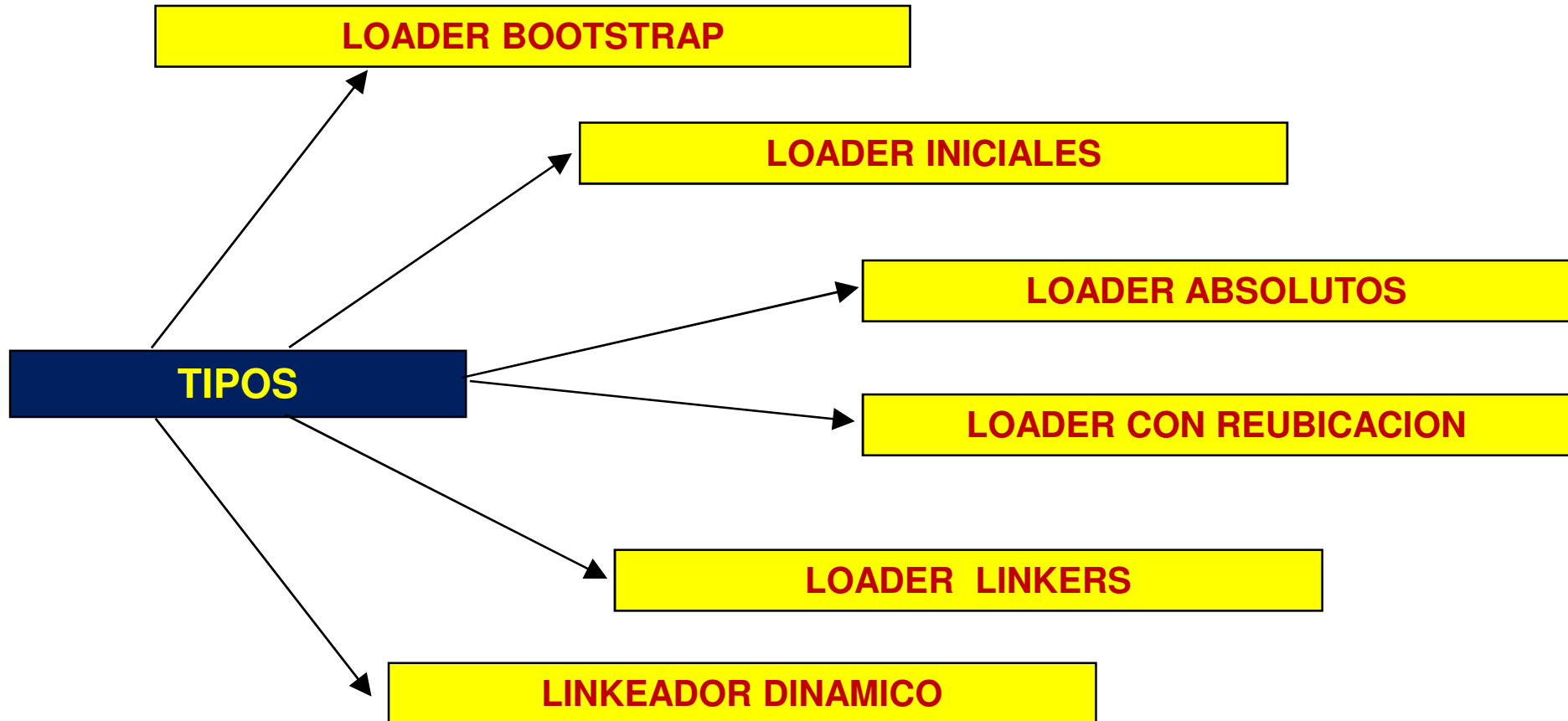
TRADUCTOR DE PROGRAMAS EN LENGUAJE ASSEMBLER A CODIGO CORRESPONDIENTE AL LENGUAJE MAQUINA



## LOADERS & LINKERS

LOADER: PRG QUE COLOCA EN MEMORIA PARA SU EJECUCION A UN PRG DE USUARIO

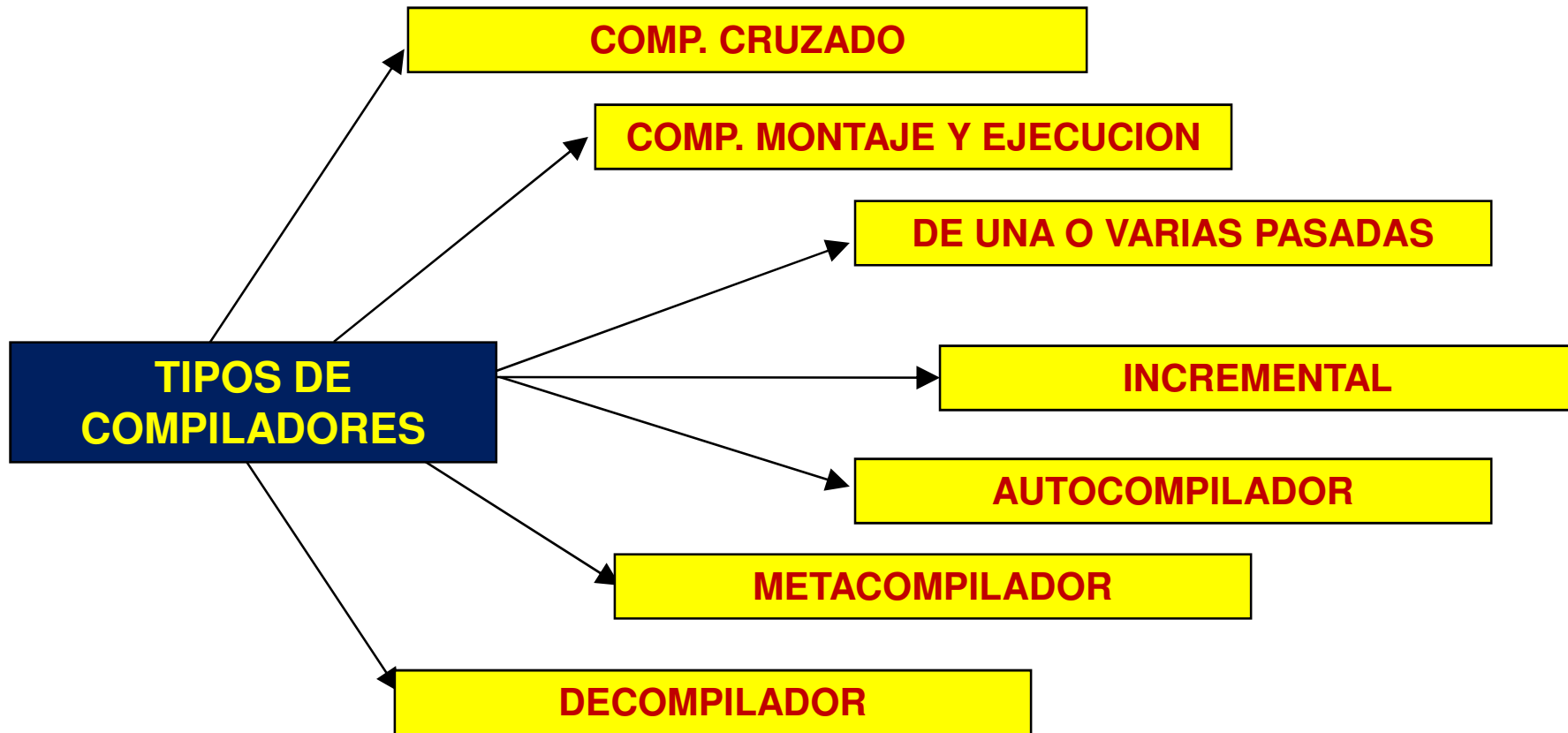
LINKER: ENLAZA PRGs OBJETO PERMITIENDO REFERENCIAS CRUZADAS ENTRE VARIABLES Y CODIGO.



## INTERPRETES Y COMPILADORES

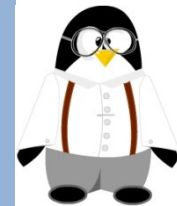
INTERPRETES: TRADUCTOR QUE REALIZA LA EJECUCION LINEA A LINEA DE UN CODIGO, CON FASES DE EDICION E EJECUCION ESTAN INTEGRADAS.

COMPILADOR: PRG QUE CONVIERTE CODIGO FUENTE EN CODIGO OBJETO, SE EJECUTA POR BLOQUE.



## BIBLIOGRAFIA DE REFERENCIA

1. Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
2. Sistemas Operativos. 5ta Ed. William Stalling. Pearson Prentice Hall. Madrid. 2006
3. Sistemas Operativos. 7ma Ed. William Stalling. Pearson Prentice Hall. Madrid. 2012
4. Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.
5. Unix, Sistema V Versión 4. Rosen,Rozinsky y Farber.McGraw Hill. NY 2000.
6. Lunix, Edición especial. Jack Tackett, David Guntery Lance Brown. Ed. Prentice Hall. 1998.
7. El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.





**Lic. en Sistemas de Información**  
**FIN UNIDAD 1**  
**INTRODUCCION Y CONCEPTOS**  
**GENERALES**

