

Computación 1

Conociendo la herramienta de cálculo

La Computadora



Sistemas Operativos ¿qué hacen?

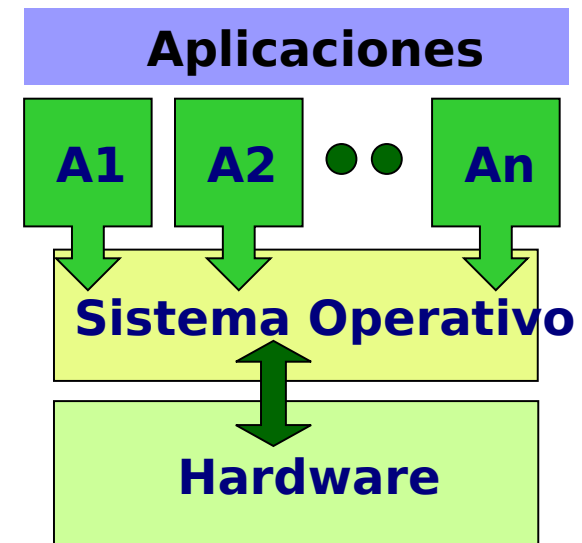
Gestionan y coordinan el uso de los recursos del equipo haciendo que los detalles particulares del hardware sean transparentes para las aplicaciones

Es un intermediario entre los programas y el hardware (recursos)

Consideramos recursos a:

- ↩ Memoria RAM
- ↩ Procesador
- ↩ Captura y despliegue de información
- ↩ Discos, disquetes, CD-ROM
- ↩ Conexión de red
- ↩ Dispositivos varios
- ↩ Etc.

■ No son imprescindibles pero ...



Sistemas Operativos Necesidad de...

¿Cómo se ejecuta un programa?

- Se lee desde algún dispositivo de memoria estable
- Se copia en la memoria RAM
- Se ordena la ejecución de la primer instrucción

Sistemas Operativos

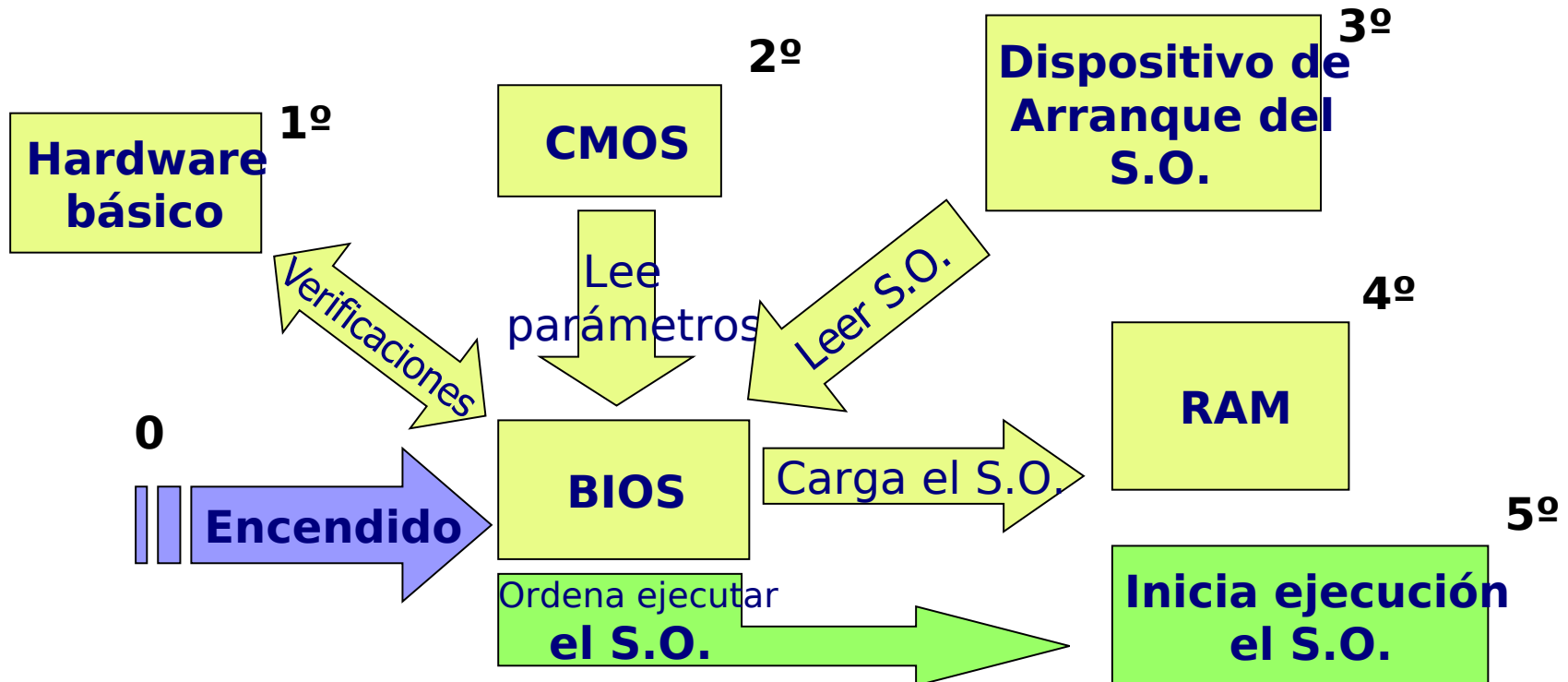
¿cómo se ejecutan? ¹

- 0- Al encender el equipo se comienza a ejecutar el programa BIOS almacenado en una EEPROM
- 1º- BIOS verifica que el hardware básico esté instalado y operativo
- 2º- BIOS lee parámetros de la instalación desde la memoria CMOS
- 3º y 4º- BIOS lee el SO según dirección dada por la CMOS y carga el núcleo de él en memoria RAM
- 5º- BIOS ordena que se ejecute la 1ª instrucción del SO

Sistemas Operativos

¿cómo llegan al control del equipo?

Proceso de arranque de un computador
(Boot process)



Sistemas Operativos

¿cómo se ejecutan? 2

Una vez que se empieza a ejecutar el SO:
Termina de cargarse a sí mismo en la memoria RAM

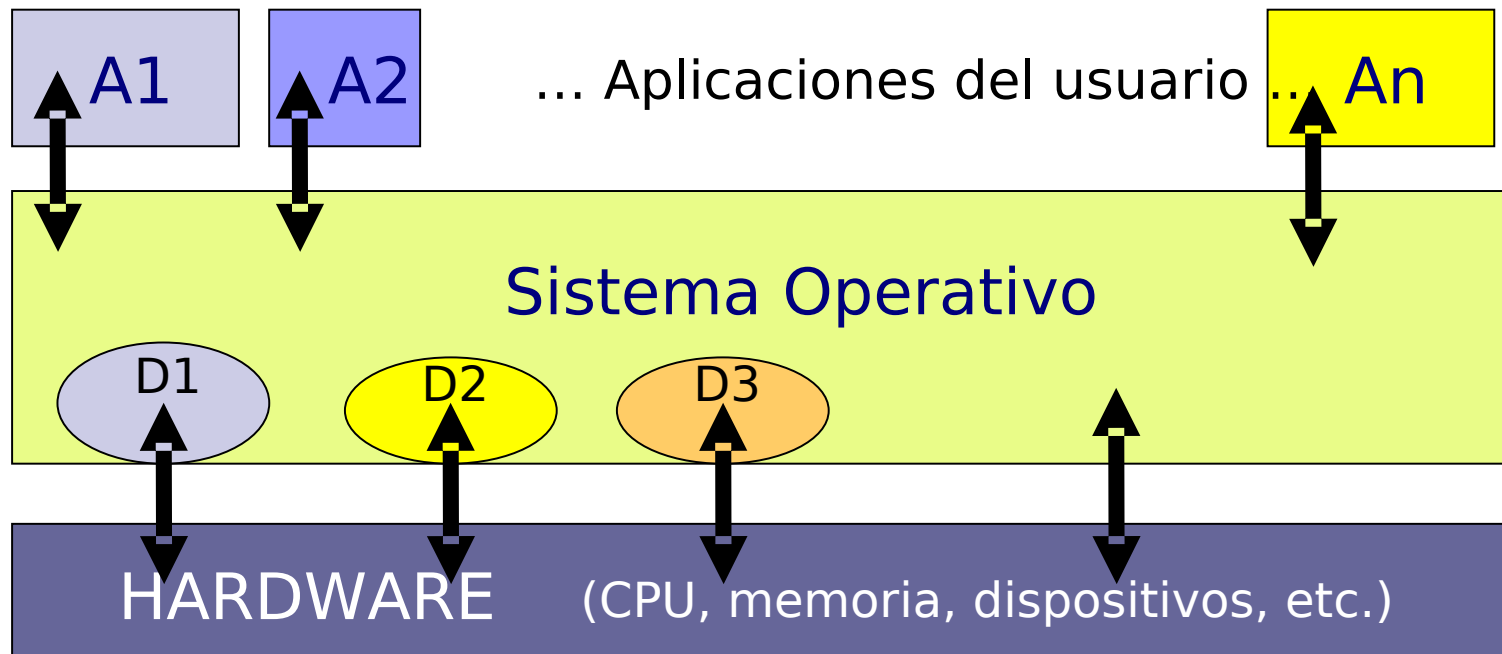
- Carga los programas de control de los diferentes dispositivos y otros programas “residentes”
- Queda a la espera de órdenes del usuario y atiende interrupciones.

Sistemas Operativos

¿cómo funcionan?

Esquema de relaciones:

aplicaciones \leftrightarrow SO \leftrightarrow hardware



Sistemas Operativos

Ejemplos PC

- Unix Berkley
- Unix System 5: Solaris, AIX, Linux
- Microsoft Windows
- Mac OS

Sistemas Operativos

Ejemplos Celulares

- Android (Google)
- iOS (Apple)
- Windows Phone (descontinuado)
- Windows 10 Mobile (Microsoft)

Sistemas Operativos

Clasificaciones

■ Cantidad de usuarios

← Monousuario

- Puede trabajar un solo usuario por vez

← Multiusuario

- Permiten trabajar en forma simultánea

■ Cantidad de tareas “concurrentes”

← Monotarea

← Multitarea

Sistemas Operativos

- Sistemas Operativos Distribuidos
- Procesamiento en paralelo
 - ← Memoria compartida (multiprocesador)
 - ← Memoria distribuida (cluster)

Sistemas Operativos

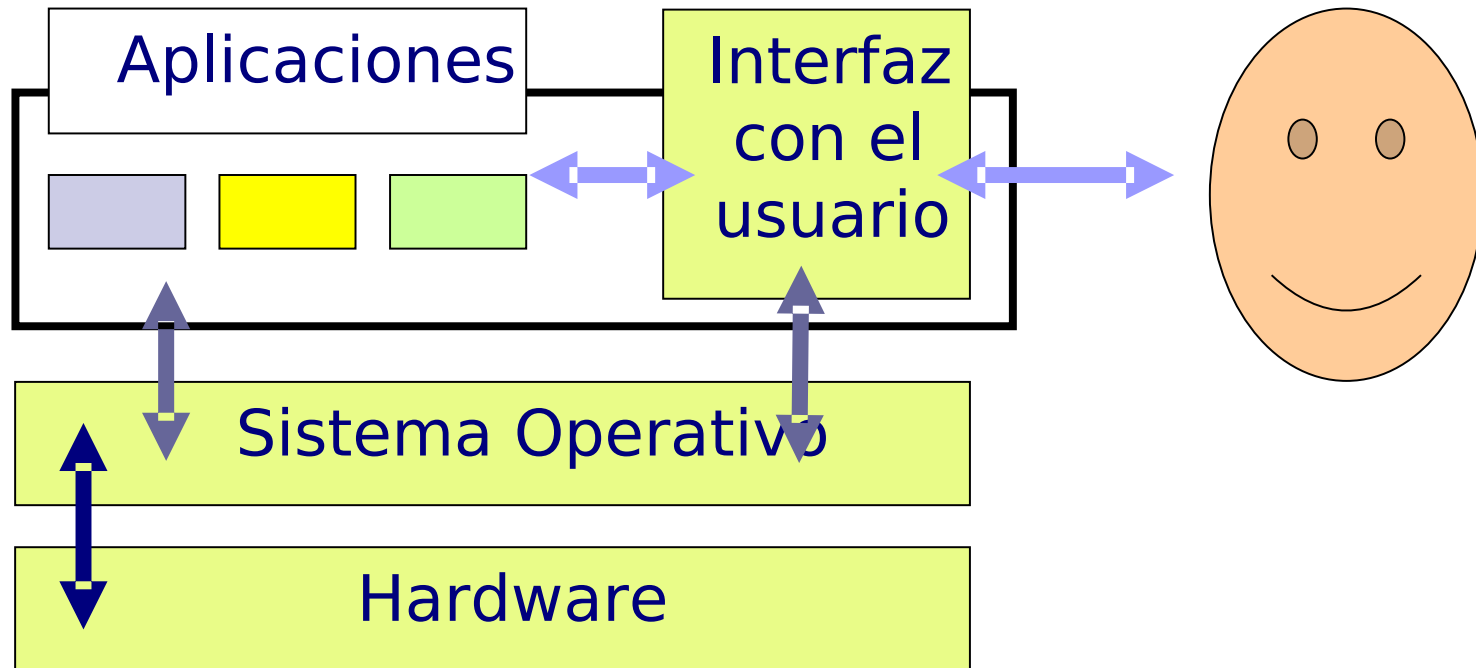
¿cómo nos comunicamos con ellos?

Interfaz hombre - máquina

- Cada SO tiene su interfaz con el usuario y eventualmente puede tener más de una
- Cada una tiene un *protocolo* de comunicación definido
- La Interfaz es una aplicación más que ejecuta el SO → NO es el SO

Sistemas Operativos

Interfaz hombre - máquina: esquema



Sistemas Operativos

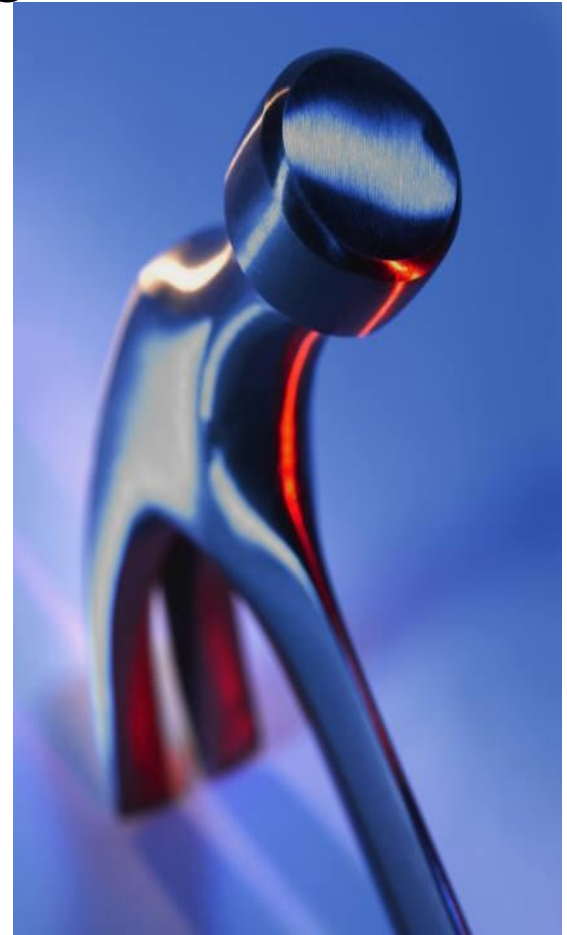
Diferentes interfaces de diálogo

- “de texto”: 80 cols x 25 líneas
 - ← DOS, cmd de Windows
 - ← Shell de Linux
- Gráfica (ventanas) Grafical User Interface
 - ← MS-Windows
 - ← Apple
 - ← Motif
 - ← Gnome, KDE, xFCE, etc.

Herramientas de software

Programas (software) de base

- Sistema Operativo
- Interfaz gráfica - GUI -
- Editores de texto
- Navegador de Internet
- Administrador de correo electrónico
- Planilla electrónica
- Manejador de Base de Datos (DBMS)



Herramientas de software

Programas de aplicación del usuario

Son aquellos que tienen una función específica para el usuario:

- Liquidación de sueldos
- Facturación
- Sistema contable
- Gestión de stock



Editor de texto plano

“texto plano” = “texto ASCII” = “texto sin formato”

Son sinónimos de un texto donde lo importante es lo que está escrito

- Hay una sola tipografía (tipo de letra)
- No hay gráficos
- No hay colores

Son ideales para suministrar información a un programa o sistema

Editor de texto plano

Ejemplos:

- MS-NotePad
- vi/vim
- Gedit / Kate (linux)
- NotePad++
- SublimeText



Procesador de texto

Texto con formato enriquecido

El texto es “enriquecido” con elementos que permiten organizar y jerarquizar la información mediante:

- Diferentes tipografías (tipos de caracteres)
- Destques: subrayado, negrita, tamaños
- Colores
- Inserción de imágenes e hipervínculos
- Es ideal para la lectura por “humanos”

Lenguajes de programación

Definición genérica de lenguaje:

“conjunto de símbolos y reglas acerca de cómo se organizan esos símbolos”.

Los procesadores de las computadoras tienen su lenguaje “de máquina” también llamado “binario”.

Un lenguaje de programación nos permite transcribir un algoritmo en un conjunto de instrucciones precisas.

Lenguajes de programación

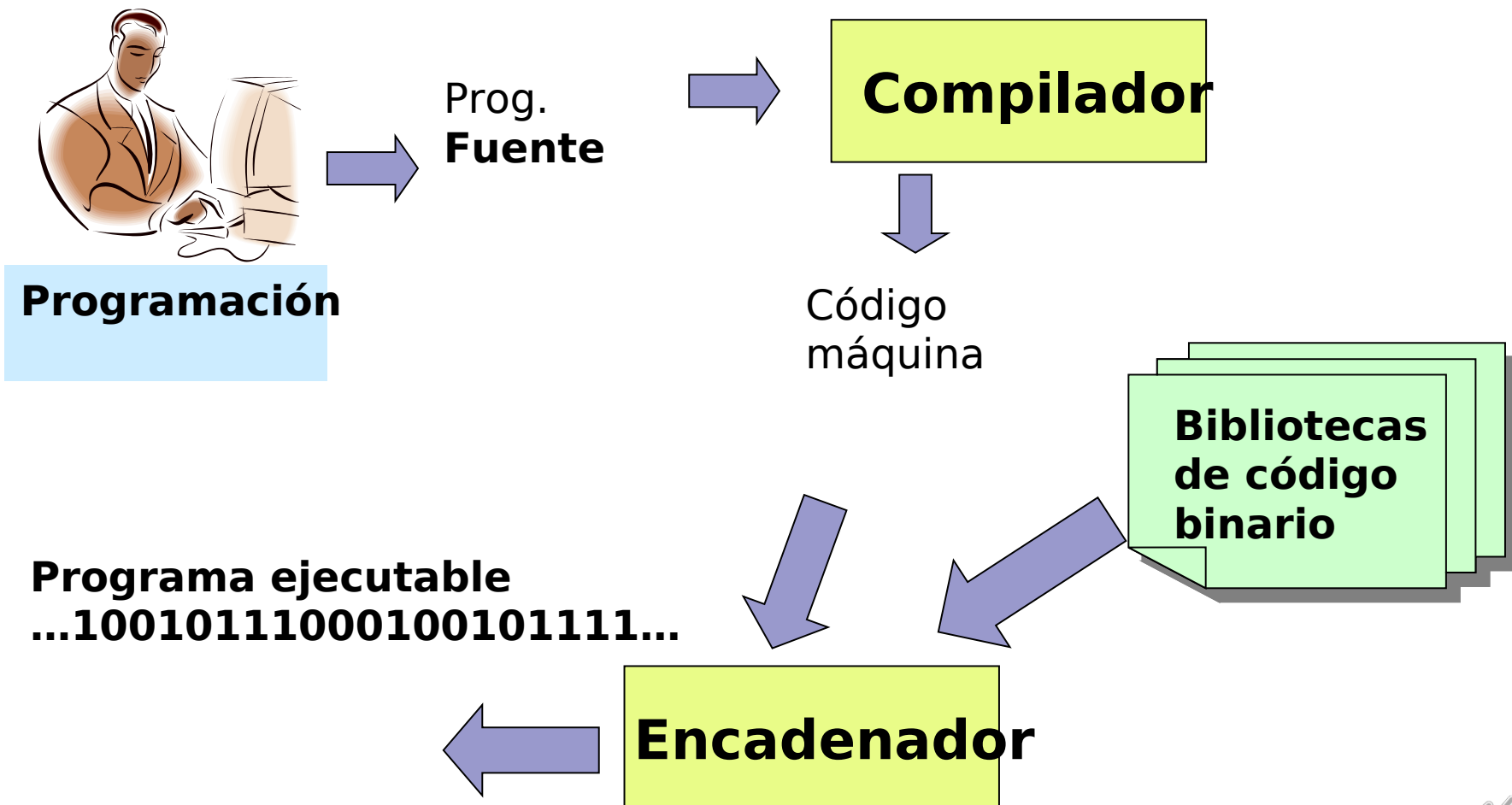
- Científicos: FORTRAN, APL, Matlab, Octave
- Comerciales
 - ← Cobol
 - ← RPG
 - ← PL/SQL (Oracle)
- Objetivos genéricos
 - ← C
 - ← C++
- Simulación: SIMULA, GAMS
- Generación de animaciones: FLASH (Macromedia)
- Otros paradigmas: Funcionales, Lógicos.

Lenguajes de programación

- Se traduce de lenguaje de programación a lenguaje de máquina, dos estrategias:
 - ← Compilar
 - ← Interpretar

Lenguajes de Programación

Compilados



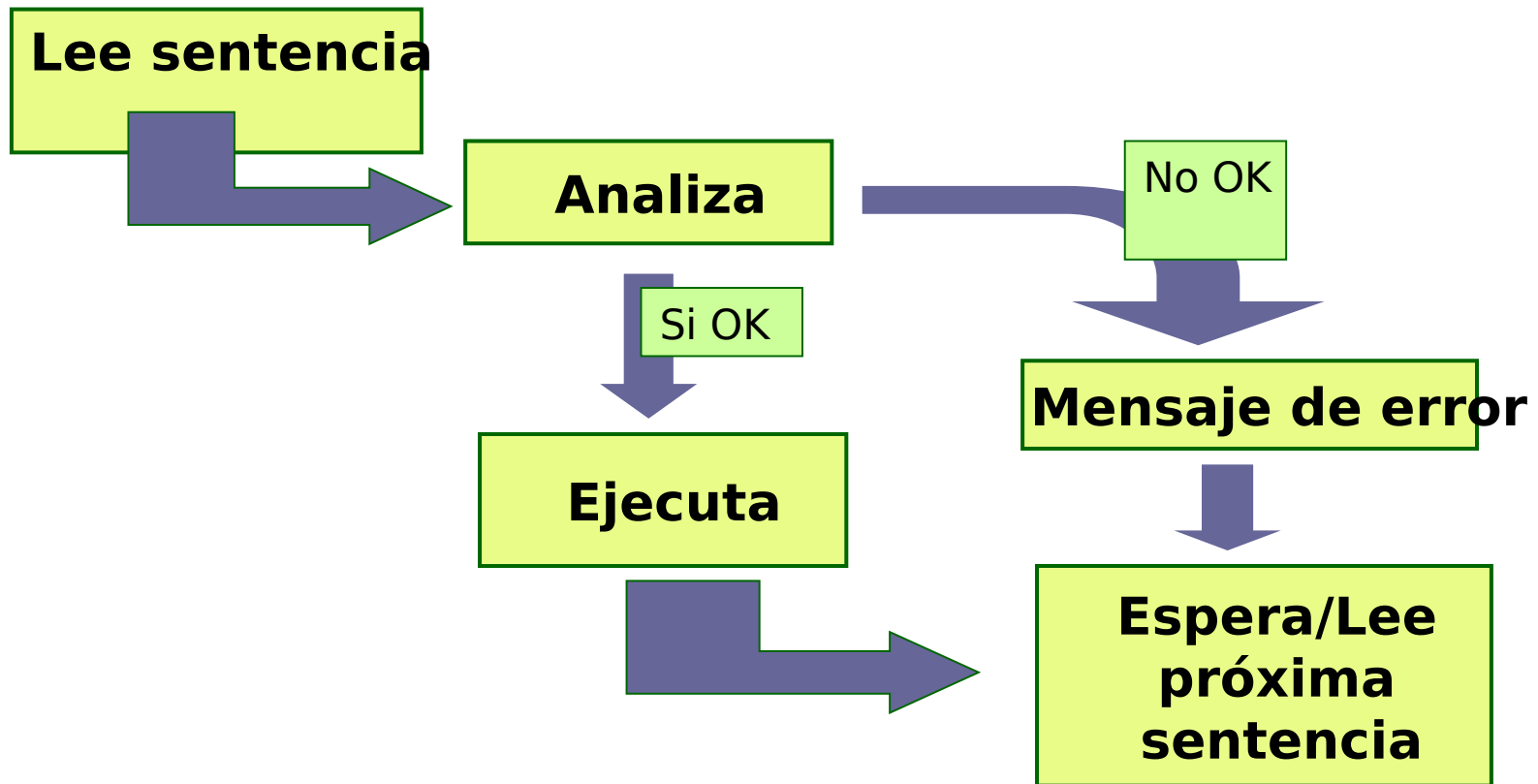
Lenguajes de programación

Compilados

- Binarios muy grandes para cargar en la memoria RAM
- Bibliotecas dinámicas
 - ⌚ DLL – Dynamic Link Library
 - ⌚ dso/so – (Dynamic) Shared Objects

Lenguajes de programación

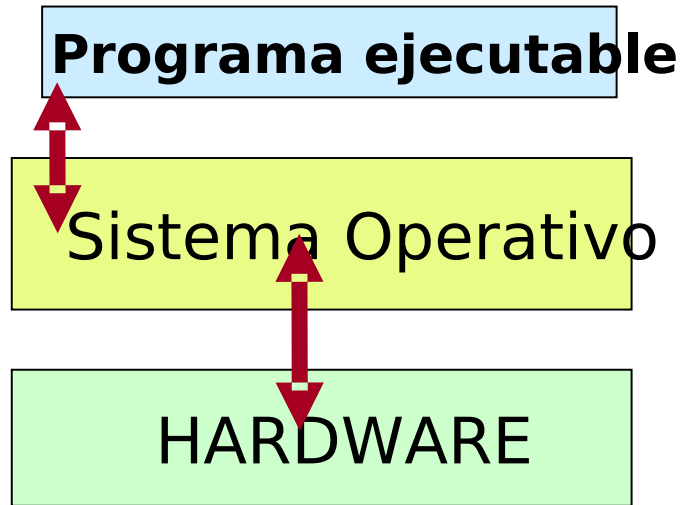
Interpretados



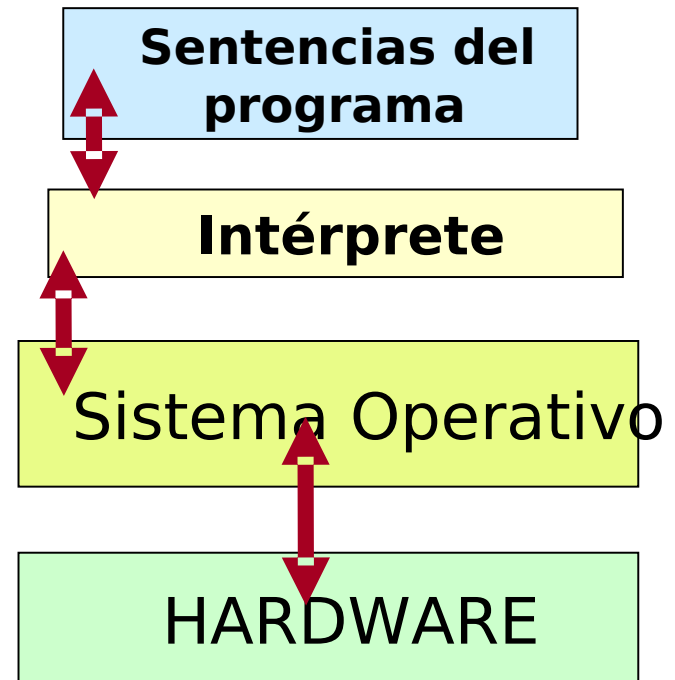
Lenguajes de programación

Compilados e Interpretados:
¿cómo se ejecutan?

Programa compilado



Programa interpretado



Lenguajes de programación

Compilados versus Interpretados

Compilados

- Más rápidos al ejecutar (se ejecutan en el procesador)
- Un código ejecutable para cada plataforma
- No es cómodo para modelar cuando no se tiene un algoritmo

Interpretados

- Más lentos al ejecutar (hay que interpretar cada vez)
- El programa es válido para cualquier plataforma con el intérprete
- Más fácil para modelar y hacer análisis de datos rápidos

Lenguajes de programación

Compilados e Interpretados: 1^{er} solución de compromiso

Compilar hasta un nivel intermedio entre “lenguaje de programación” y “lenguaje de máquina”

Se economiza tiempo pues no se repite:

- Análisis sintáctico
- Consistencia de uso de variables
- Consistencia de uso de estructuras

El código resultante se interpreta

Lenguajes de programación

Compilados e Interpretados: 2^a solución de compromiso – JIT compilers -

■ **Si** el fuente es más nuevo que la versión compilada en el “cache”

■ **Entonces** se compila generando “Código Binario”

■ **FinSi**

El “Código Binario” resultante se ejecuta

Lenguajes de programación

■ Java

- ⊖ Un solo lenguaje de programación
- ⊖ Máquina virtual Java (JVM)

■ .Net

- ⊖ Varios lenguajes de programación
- ⊖ CLR (Common Language Runtime)