



UNIVERSIDAD
DE GRANADA

Tema 1: Sistema de Cómputo

Fundamentos del Software

1º Grado en Informática

Rosana Montes - rosana@ugr.es

Objetivos

- ♦ Explicar la relación entre hardware y software
- ♦ Conocer cómo se representa la información que manipula un ordenador.
- ♦ Entender el concepto de programa y los engranajes que activa la ejecución de un conjunto de instrucciones.
- ♦ Describir las tres categorías fundamentales de software y sus relaciones.
- ♦ Esbozar la evolución de las interfaces de usuario.
- ♦ Analizar qué aplicación es necesaria en las distintas disciplinas de un profesional digital.

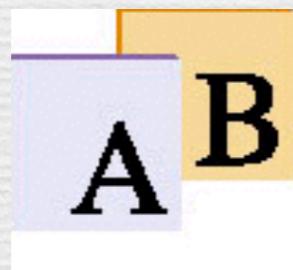
1.1 Componentes de un Sistema de Cómputo

Contenidos a tratar:

- ♦ Definiciones básicas: Informática, Ordenador
- ♦ Unidades de medida: bit, Byte y múltiplos.
- ♦ Tratamiento automático de la información.
- ♦ Representación interna de la información.
- ♦ Códigos intermedios. Eficiencia de un código.
- ♦ Dato, Instrucción, Programa, Lenguaje de programación alto nivel, Lenguaje máquina.
- ♦ Sistema Informático. Relación hardware y software.
Firmware.
- ♦ Clasificación del hardware y del software.

Definiciones Básicas [Prie06] (pp.1-7)

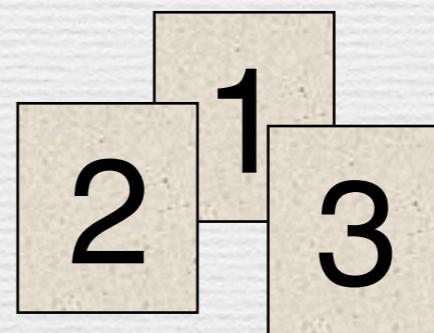
- Informática “conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información mediante ordenadores”.
- INFORmación y autoMÁTICA**
- Ordenador “máquina electrónica dotada de una memoria de gran capacidad y métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización de programas”
 - Información “elementos que procesa un ordenador”



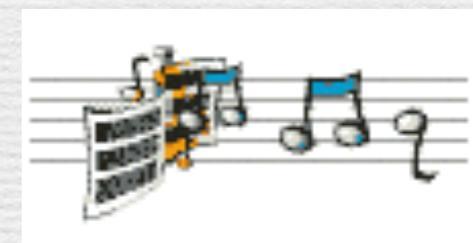
Texto



Imágenes



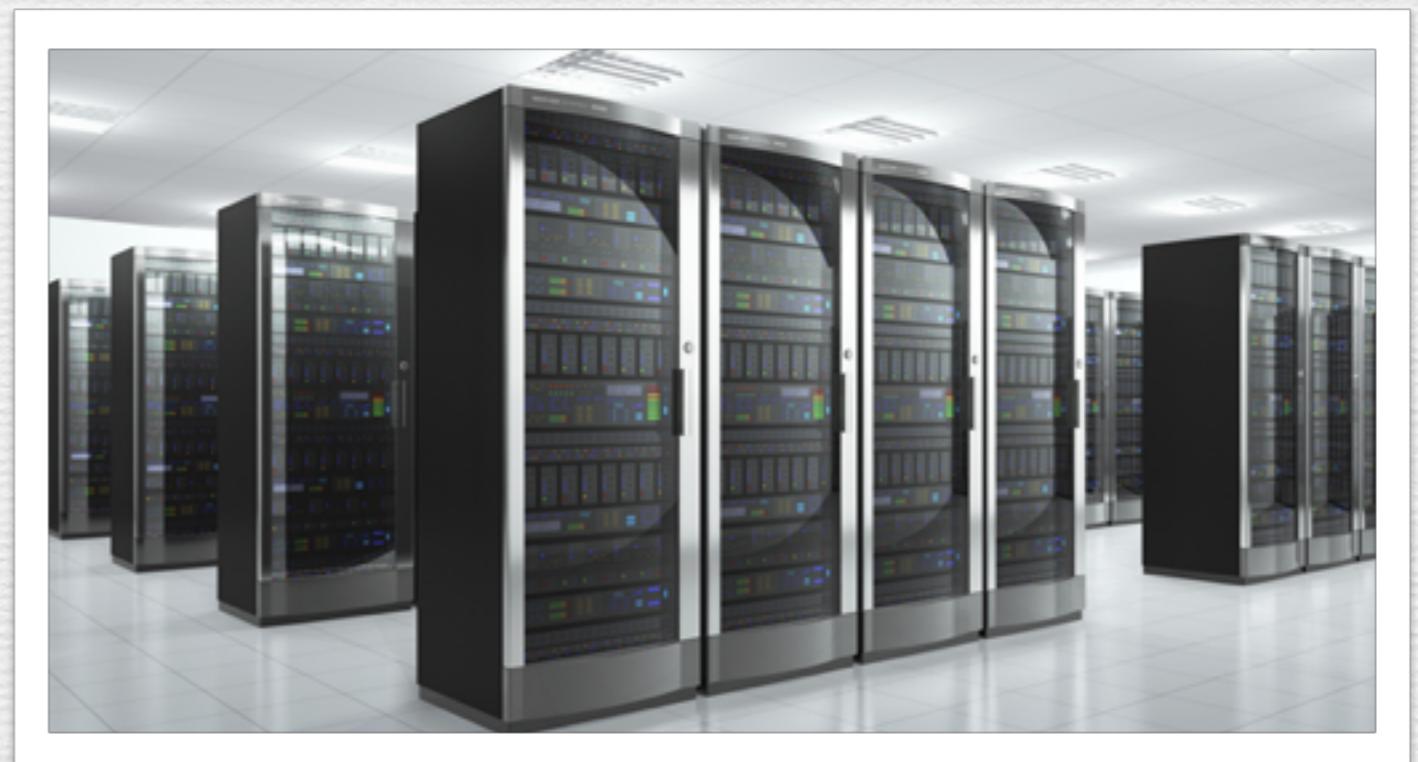
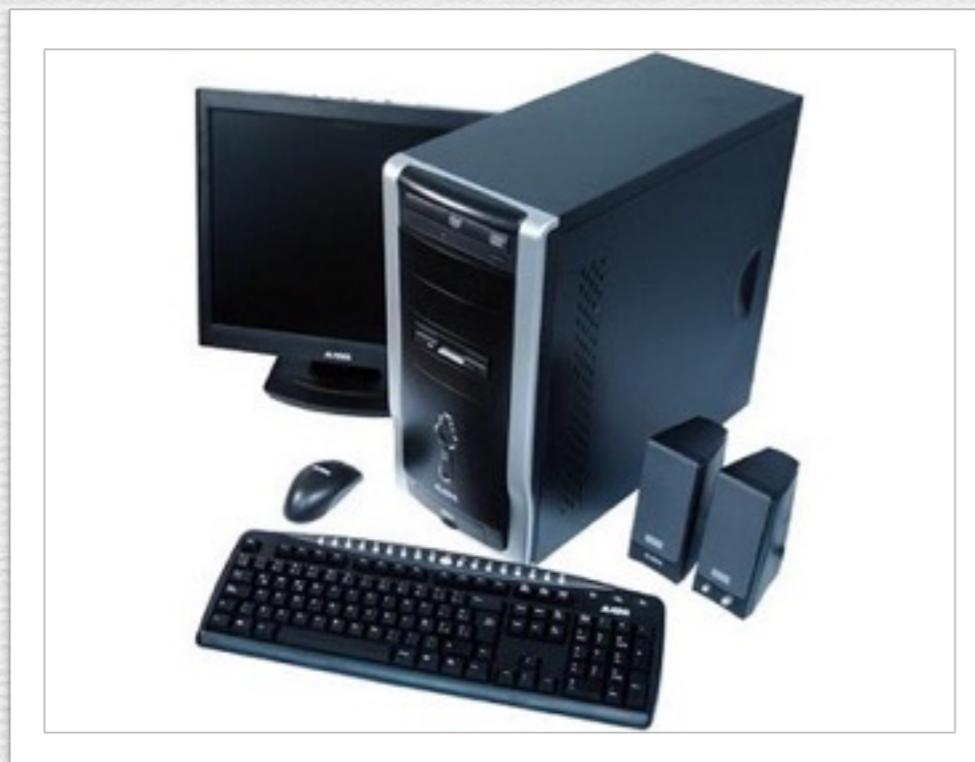
Números



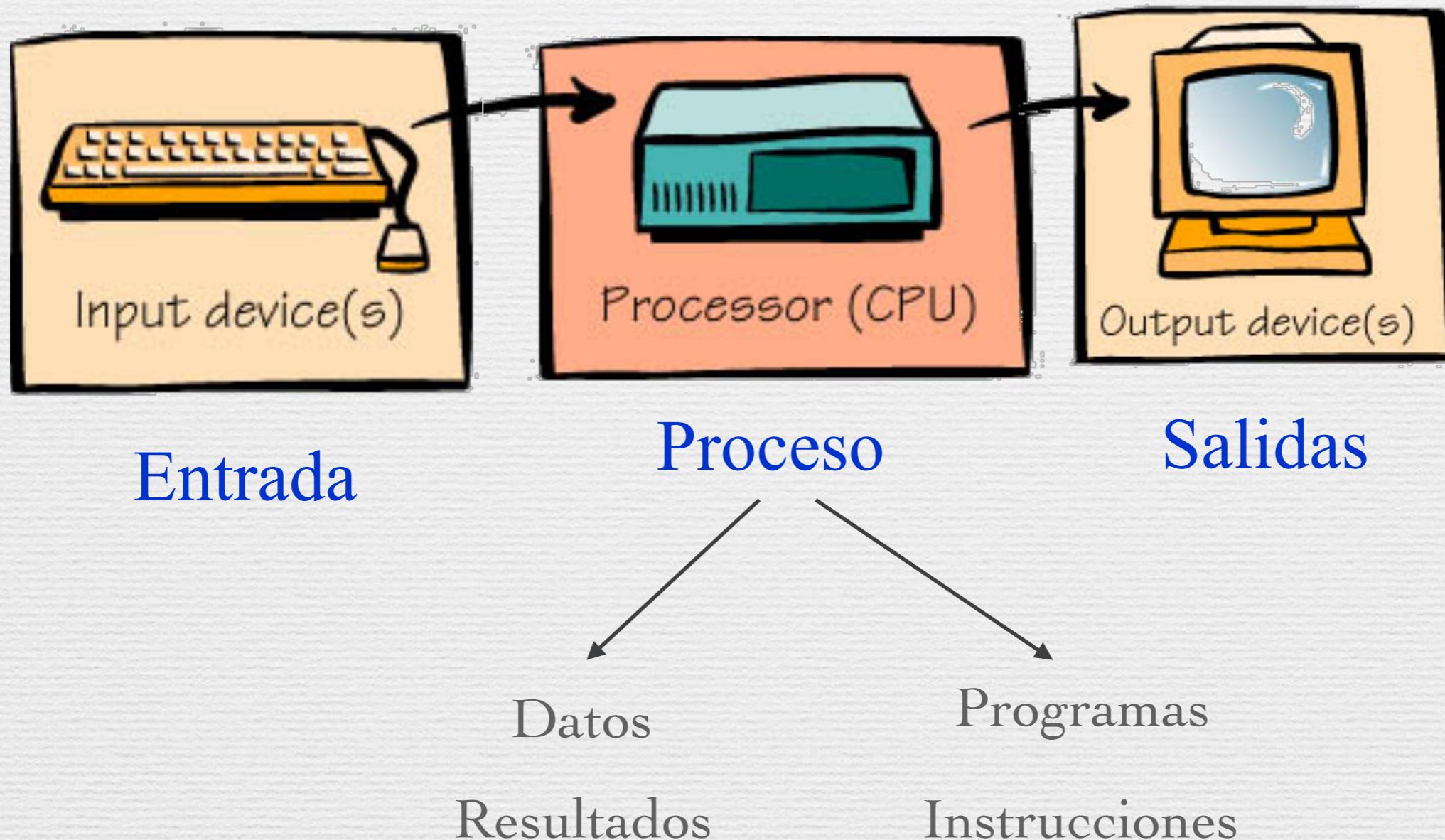
Sonidos

Ordenador =

Dispositivo físico que admite datos a través de sus elementos de entrada, los transforma al ejecutar un programa almacenado y envía los resultados a través de sus elementos de salida.



Tratamiento Automático de la Información



Unidades de Medida

▪ Byte	= 8 bits
▪ Kilobyte (KB)	= 2^{10} = 1024 Bytes
▪ Megabytes (MB)	= 2^{20} Bytes
▪ Gigabytes (GB)	= 2^{30} Bytes
▪ Terabytes (TB)	= 2^{40} Bytes

- 1 Kilobit (Kb) = 2^{10} b
- 1 Megabit (Mb) = 2^{10} Kb
- 1 Gigabit (Gb) = 2^{10} Mb
- 1 Terabit (Tb) = 2^{10} Gb
- 1 Petabit (Pb) = 2^{10} Tb

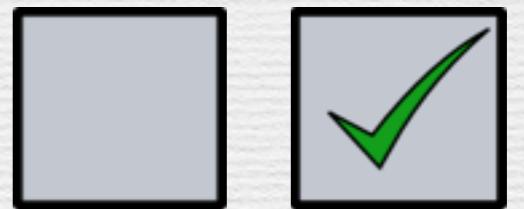
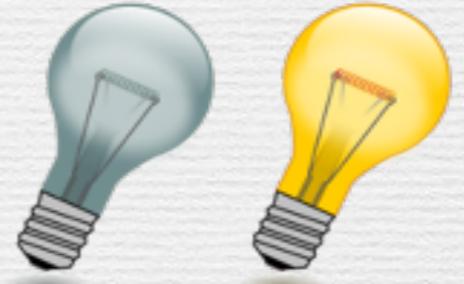
Fundamentos de los bits [Beekman06] (pp.65-70)

Unidades de medidas

Bit	Es el elemento más pequeño de información. Un bit es un único dígito en un número binario (0 o 1)
Byte	Unidad básica de almacenamiento de información. Equivalente a 8 bits.
Kilobyte	Equivale a 1024 bytes. Unidad común para indicar tamaño de ficheros de texto.
Megabyte	Unidad de medida de cantidad de datos informáticos. Equivale a 1024 KB. Como múltiplo binario del byte que equivale a un millón de bytes (1 048576 bytes)
Gigabyte	Unidad común para indicar el tamaño de la memoria principal, así como del almacenamiento permanente. Un gigabyte equivale a 1024 MB. Con exactitud es 1,073,742,824 bytes.
Terabyte	Es la unidad de medida de la capacidad de memoria y de dispositivos de almacenamiento informático. Su símbolo es TB y coincide con algo más de un trillón de bytes. Equivale a 1024 GB.

Información

- Bit: unidad mínima de información
- Codifica información:
 - 1 bit: 0 ó 1
 - 2 bits: 00, 01, 10 ó 11
 - ... a mayor número de bits mayores cantidades de información se codifican



0 1

n = número de bits para codificar

False True

m = número de elementos codificados

$$m=2^n \quad n \geq \log_2(m) = 3.32 * \log_{10}(m)$$

Representación interna

- ♦ **Sistemas de numeración:** Conjunto de dígitos que nos sirven para representar ciertas cantidades
- ♦ Todo número se expresa por un conjunto de cifras cada una tiene un valor que depende de:
 - La cifra en sí.
 - De la posición que ocupa dentro del número.
- ♦ **Sistemas de representación:** decimal, binario, octal, hexadecimal, romanos, base 3, etc.

Representación interna

- ♦ Transformaciones de representación
 - Ejemplo: el número seis se podría representar de varias formas:
 - Sistema decimal: 6
 - Sistema binario: 110
 - Romanos: VI

Cambio de base: binario, octal, hexadecimal [Prie06] (Apéndice A. pp.767)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Binario (N_2)	000	001	010	011	100	101	110	111									
Octal (N_8)	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17	20
Decimal (N_{10})	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Hexadecimal	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10

- Suma de la base elevado al exponente de la posición del dígito

$$N^{\circ} \text{base } 10 = \text{digito}_0 \times \text{base}^0 + \text{digito}_1 \times \text{base}^1 + \dots + \text{digito}_n \times \text{base}^n$$

Para obtener el número en otra base: dividir hasta llegar a 1 y tomar los restos y el último cociente de la división.

$$m=2^n \quad n \geq \log_2(m) = 3.32 * \log_{10}(m)$$

Representación numérica

$$\begin{array}{r} 23 | 2 \\ \hline 1 \ 11 | 2 \\ \quad 1 \quad 5 | 2 \\ \quad \quad 1 \ 2 | 2 \\ \quad \quad \quad 0 \ 1 \end{array}$$

$$23_{(10)} = 10111_{(2)}$$

Posición 4 3 2 1 0
1 0 1 1 1

$$\begin{aligned} 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 &= \\ 1 + 2 + 4 + 0 + 16 &= \\ 23 \end{aligned}$$

Representación de los caracteres

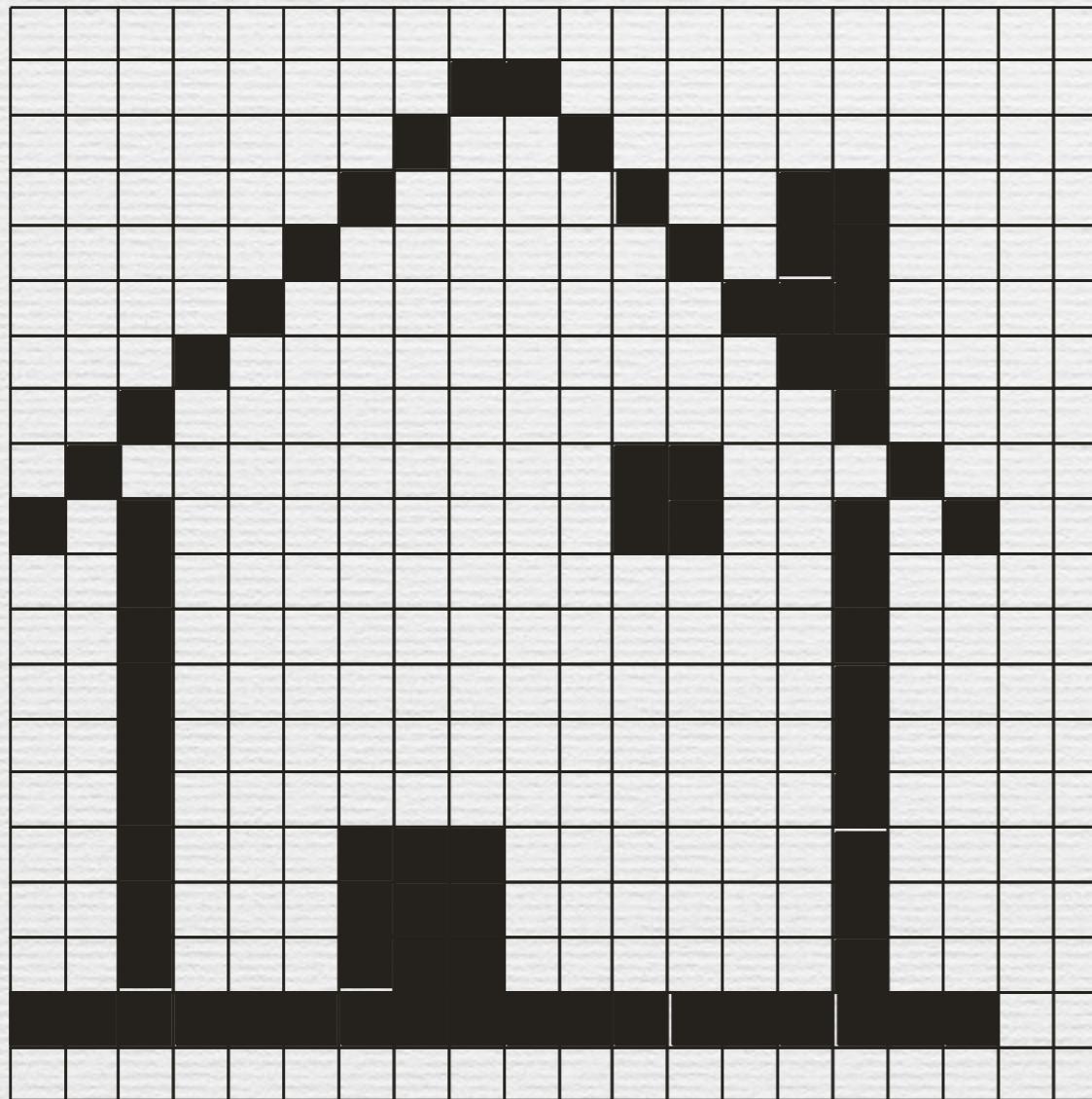
- Alfabeto: compuesto por caracteres alfabéticos, numéricos, especiales y de control.
- ASCII – American Standard Code for Information Interchange, con m=128.
- ASCII extendido con m=256. Es un código muy utilizado y representa cada carácter de forma única con 8 bits,
- Unicode - Internacionalización. UFT8, UTF16,...

Character	UTF-16	UTF-8	UCS-2
A	0041	41	0041
c	0063	63	0063
Ö	00F6	C3 B6	00F6
¾	4E9C	E4 BA 9C	4E9C
♪	D834 DD1E	F0 9D 84 9E	N/A

Character	ASCII binary code
A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
J	01001010
K	01001011
L	01001100
M	01001101
N	01001110
O	01001111
P	01010000
Q	01010001
R	01010010
S	01010011
T	01010100
U	01010101
V	01010110
W	01010111
X	01011000
Y	01011001
Z	01011010
0	00110000
1	00110001
2	00110010
3	00110011
4	00110100
5	00110101
6	00110110
7	00110111
8	00111000
9	00111001

Fuente:

Representación de la imagen



```
00000000000000000000  
000000001000000000000  
00000001001000000000  
00000010000100110000  
00000100000010110000  
00001000000001110000  
0010000000000110000  
001000000000000010000  
0100000000010001000  
10100000000010010100  
001000000000000010000  
001000000000000010000  
001000000000000010000  
001000000000000010000  
0010001100000010000  
0010001100000010000  
0010001100000010000  
11111111111111111100  
0000000000000000000000
```

Imagen raster != Imagen vectorial

Codificación

- ♦ **Eficiencia de un código (T).** Representa la bondad de un código

$$T = m / 2^n \text{ con } 0 \leq T \leq 1$$

- ♦ **Redundancia de un código (R).** Mide la inversa de la eficiencia y se expresa en tanto por ciento.

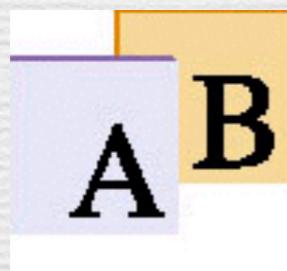
$$R = (1-T) * 100$$

- ♦ Redundancias deliberadas para detectar errores en la transmisión de información.

- **Bit de paridad**
 - Criterio par
 - Criterio impar

Datos

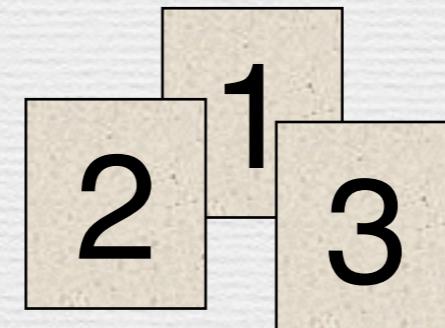
- **Instrucción:** conjunto de símbolos insertados en una secuencia estructurada o específica que el procesador interpreta y ejecuta.
- **Datos:** Símbolos que representan hechos, condiciones, situaciones o valores. Elementos de información.



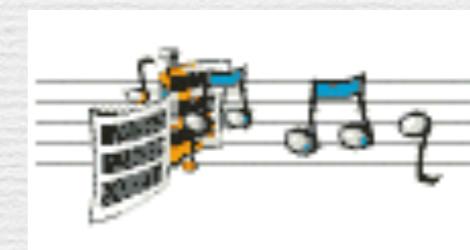
Texto



Imágenes



Números



Sonidos

Programa

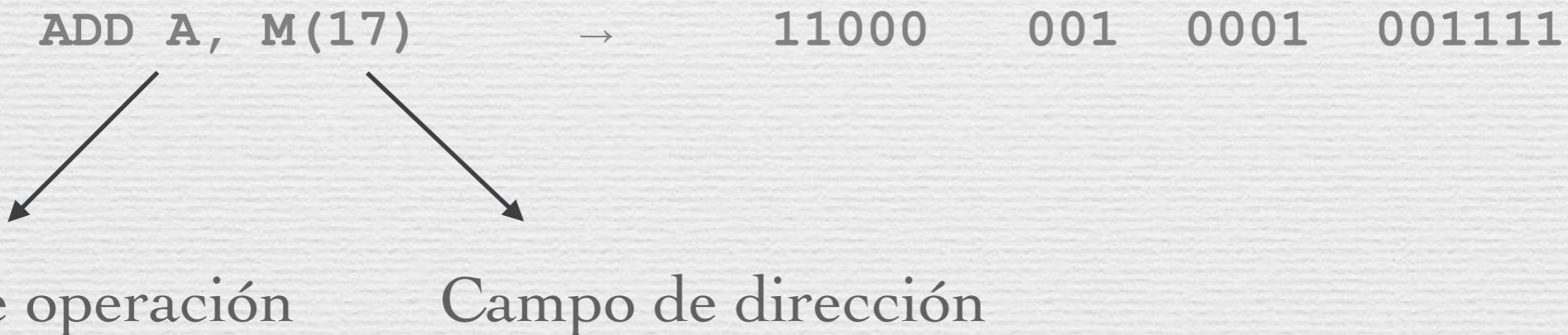
- Lenguaje natural:

Suma lo que hay en A con lo que tiene la posición 17 de una secuencia de valores.

- Lenguaje de programación de alto nivel:

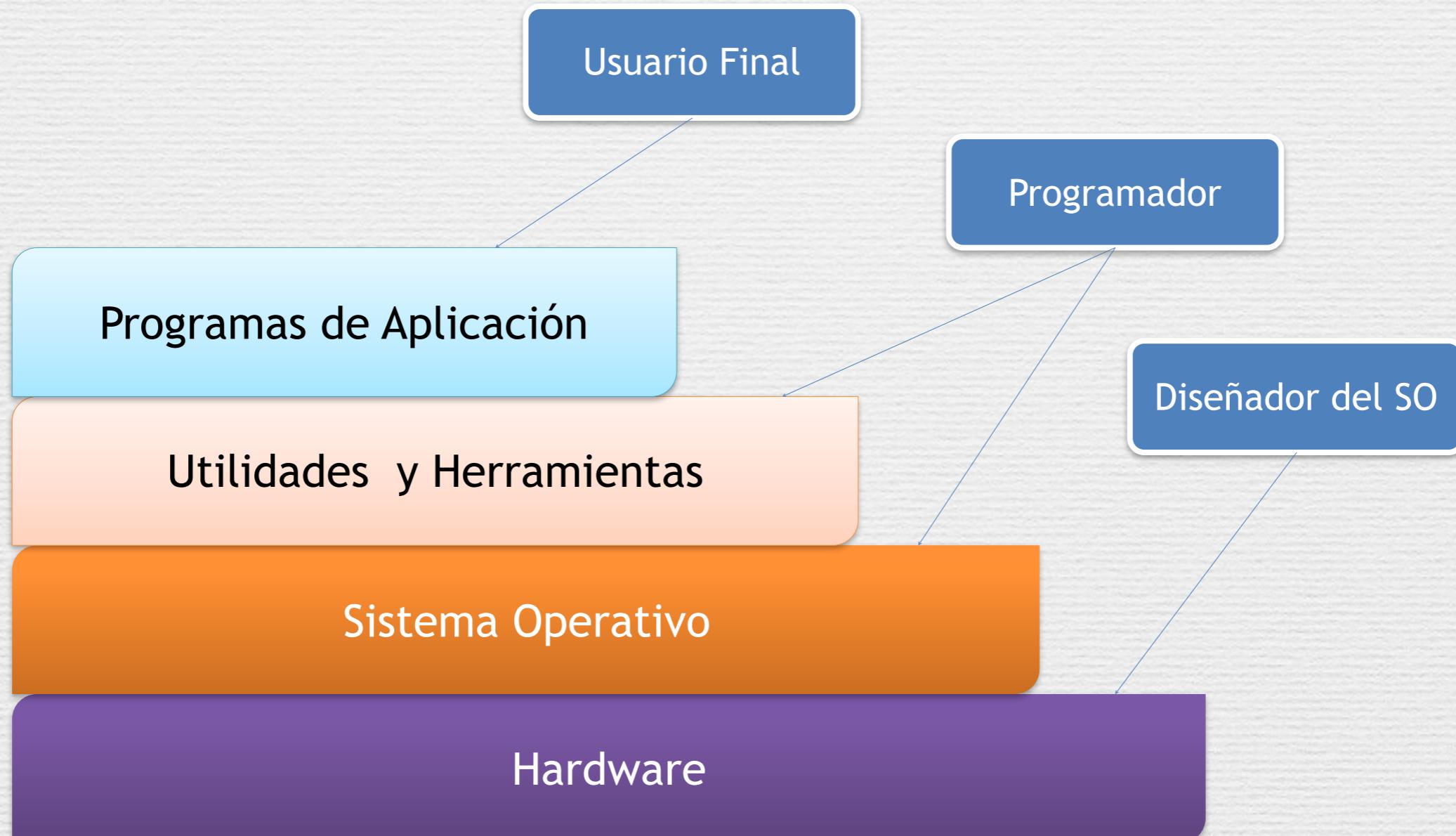
$A = A + M[17]$

- Ensamblador y lenguaje máquina:



Sistema Informático

Involucra y equilibra tres componentes principales: el hardware, el software y los propios recursos humanos.



Elementos Básicos [Stal05] (pp.55) [Stal12] (pp.28)

Hardware

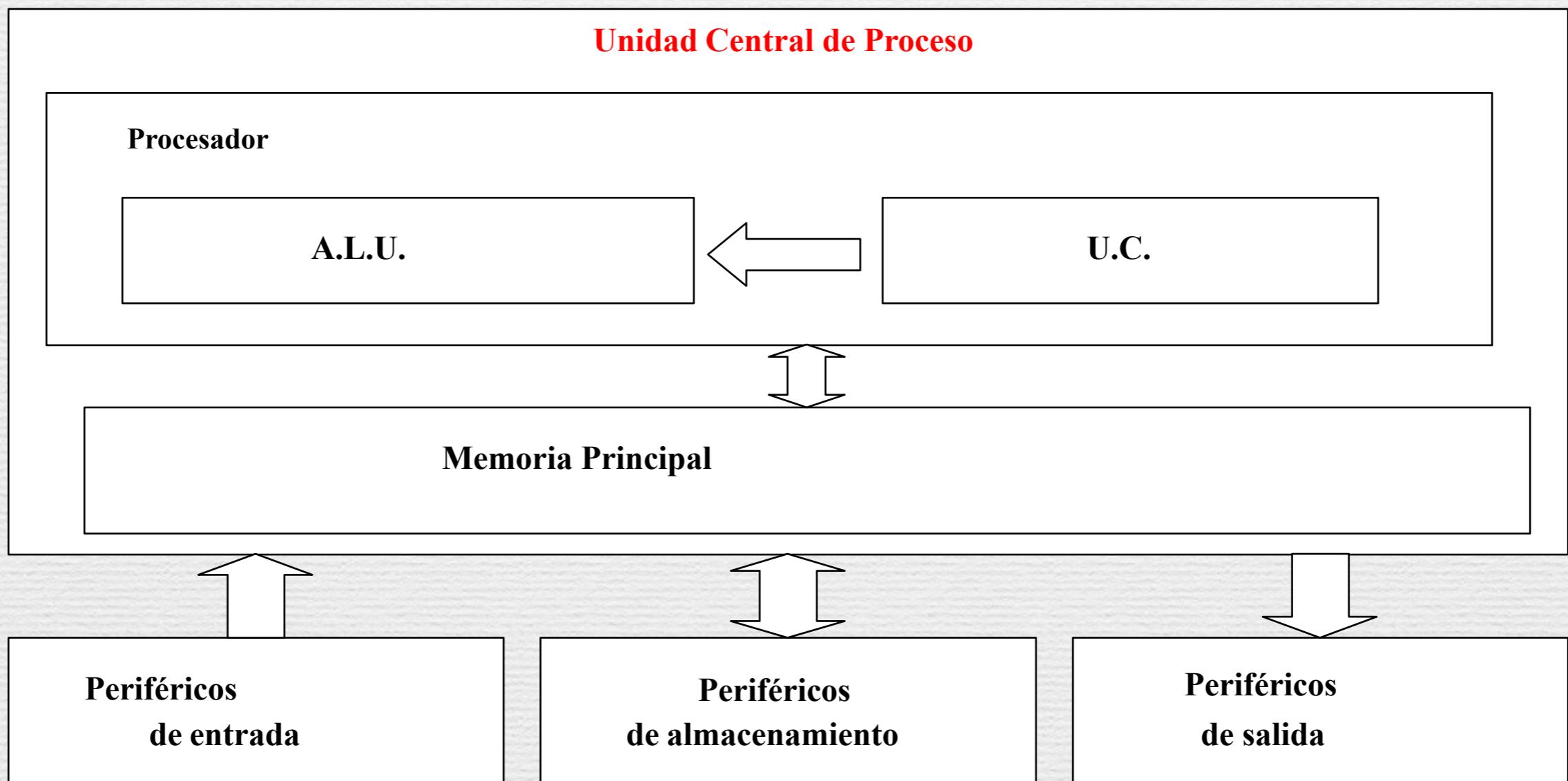


Firmware

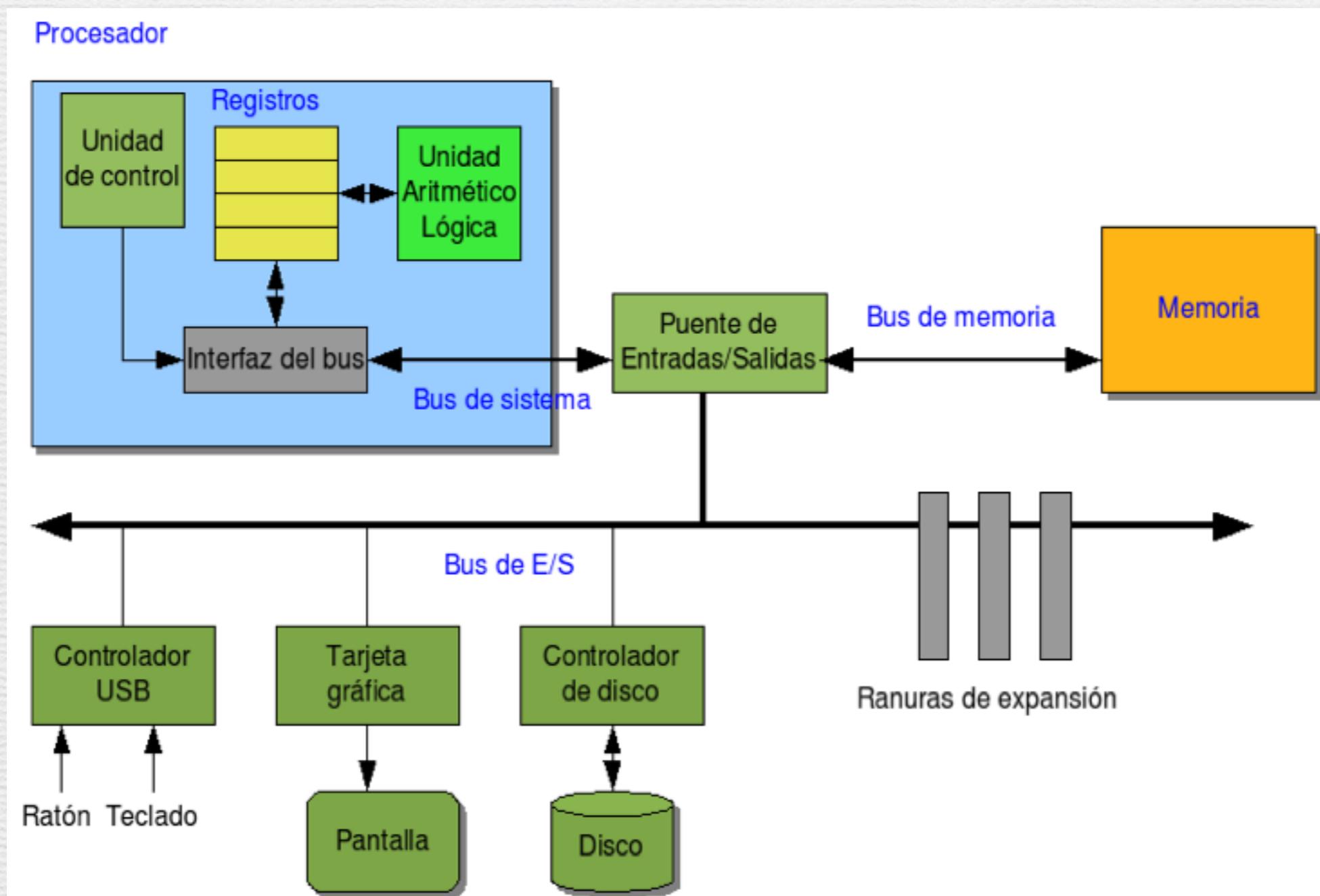


- Es parte del hardware de los dispositivos
- Es también software (microcódigo) que viene de fábrica.

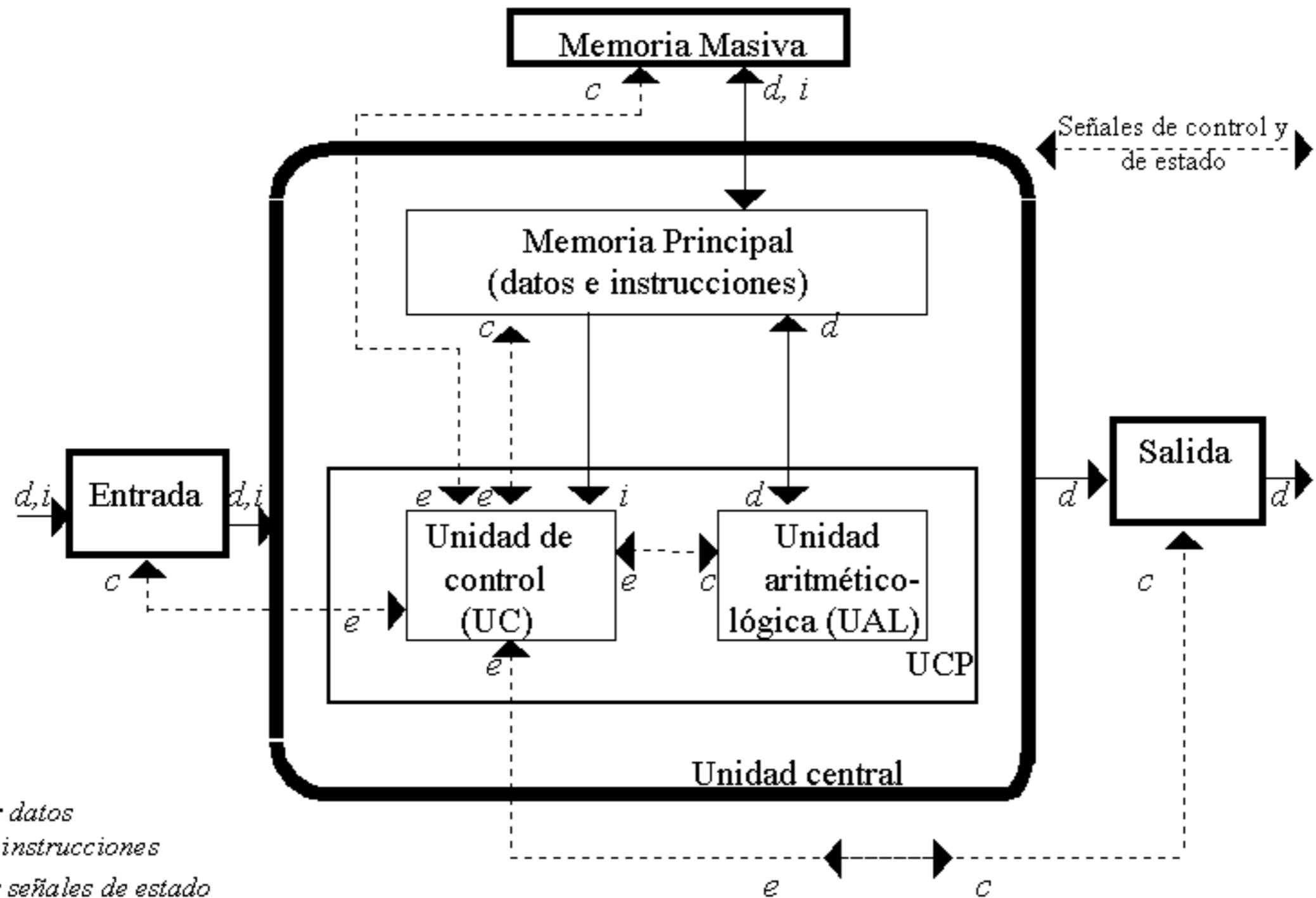
Esquema funcional de un ordenador (I)



Esquema funcional de un ordenador (II)



Esquema funcional de un ordenador (III)



Procesador

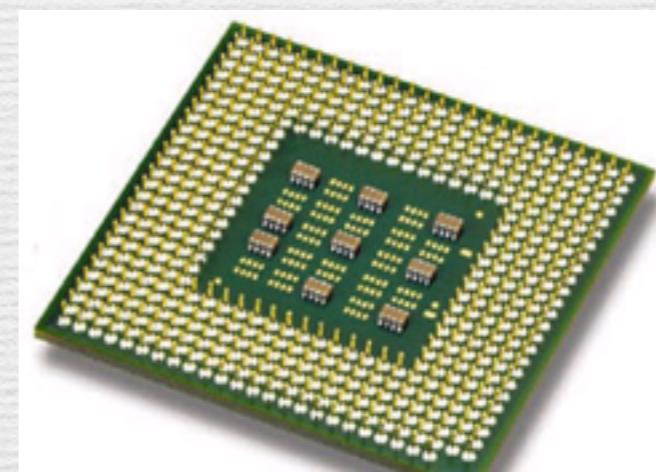
Funciones:

- Interpreta y ejecuta instrucciones (UC).
- Realiza cálculos aritméticos y lógicos sobre los datos (ALU).
- Se comunica con otras partes del sistema para gobernar el ordenador (UC).

Está compuesto por:

- Unidad de Control (UC).
- Unidad Aritmético Lógica (ALU).

Cuando todos esos circuitos se ensamblan en un único chip o pastilla, se le denomina



La Unidad de Control

Interpreta y ejecuta instrucciones.

- Detecta señales de estado
- Capta de la memoria las instrucciones
- Genera señales de control

Se comunica con otras partes del sistema para controlar el ordenador.

Elementos:

- Registro contador de programa
- Registro de instrucción
- Decodificador
- Secuenciador
- Reloj

La Unidad de Control

- La velocidad del ordenador viene determinada por el reloj o generador de pulsos de la UC, que produce señales eléctricas o pulsos que marcan los pasos de cada acción.
- Tiempo de ciclo: periodo de la señal.
- La velocidad del reloj se mide en GigaHertzs (GHz), siendo equivalente a mil millones de ciclos por segundo (10⁹)
- Existen otras unidades de medida como las Mips (millones de instrucciones por segundo)

La Unidad Aritmético Lógica

Realiza cálculos aritméticos y lógicos sobre los datos.

Utiliza registros (pequeñas memorias de gran velocidad) para almacenar resultados inmediatos.

Elementos fundamentales:

- Circuito operacional
- Banco de registros
- Registro acumulador
- Registro de estado

Memoria Principal

Función:

- Se utiliza para almacenar las instrucciones de los programas para ser ejecutados.
- Almacena los datos usados por los programas.
- Almacena resultados intermedios.

Tipos: RAM / ROM

Estructura:

- Formada por celdas (posiciones de memoria)
- Identificación de la celda (dirección de memoria)

El corazón de la computadora: CPU y memoria [Beekman06] (pp.71-78)

Memoria Principal

- ♦ **Características RAM:**

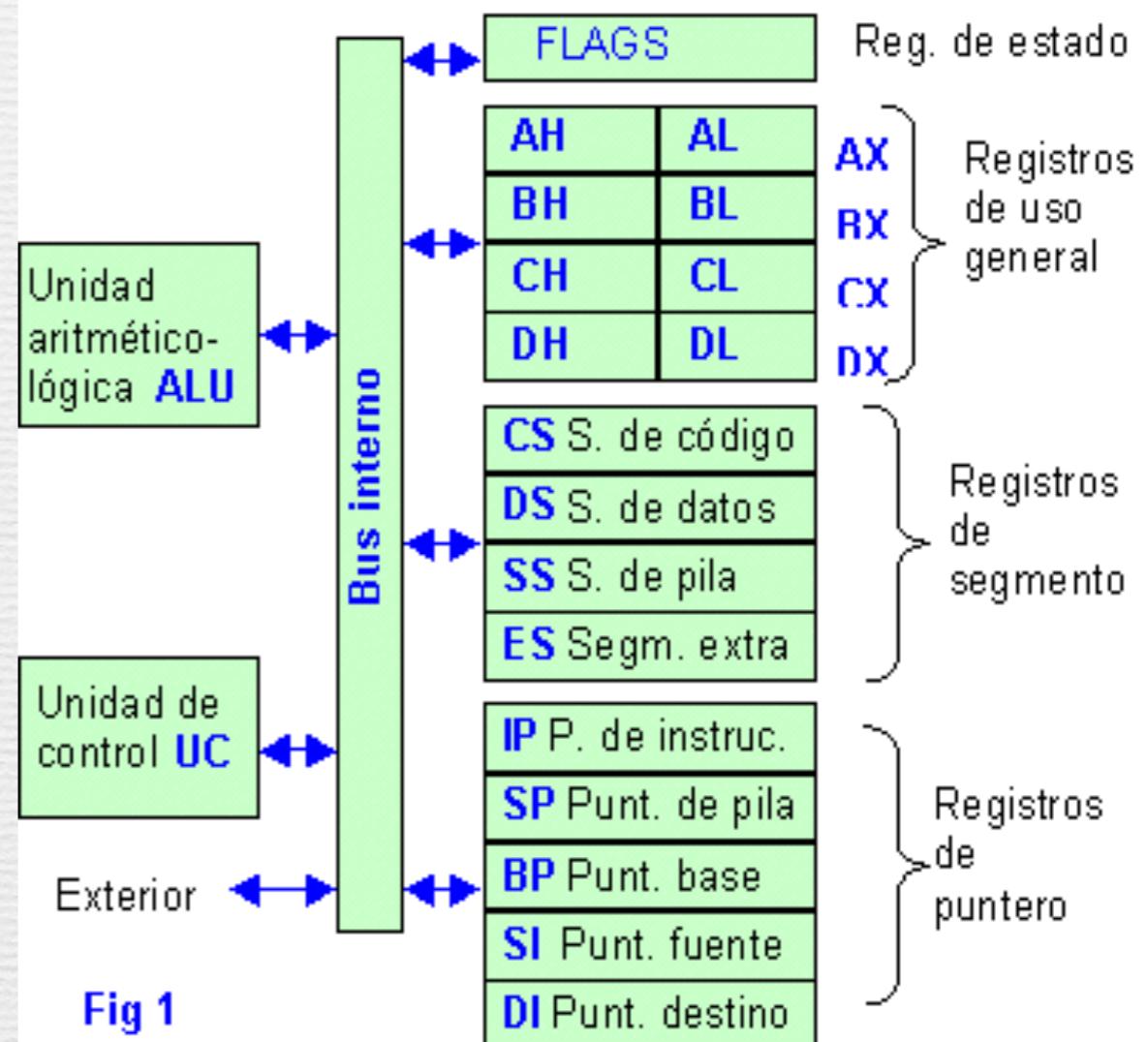
- De acceso directo
 - Dirección única para cada posición o palabra de memoria
 - La información es recuperada muy rápidamente
- Permite lectura (no destructiva) y escritura (destructiva)
- Volátil: Sin suministro eléctrico se pierden los datos

Piramide de memoria [Stal05] (pp.11-13) [Stal12] (pp.44-47)

1.2 Capa Hardware

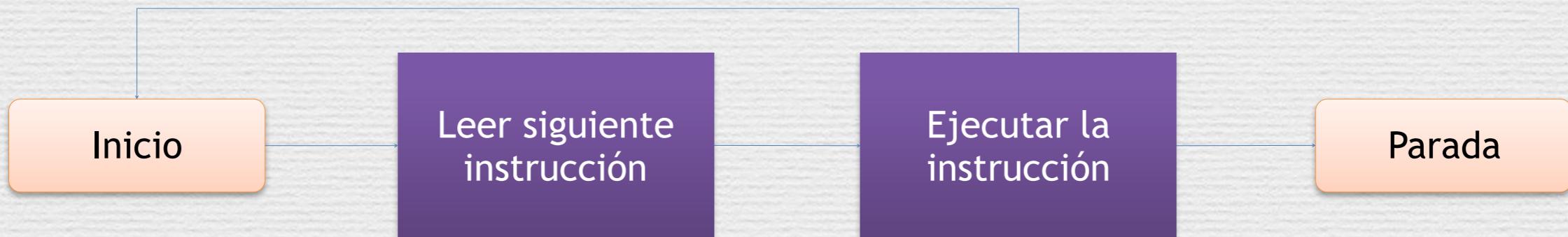
- Registros visibles para el usuario.
- Registros de control y estado
 - Contador de programa (PC).
 - Puntero de pila (SP).
 - Registro de instrucción (IR).
 - Registro de estado (bits informativos).

Esquema del microporcesador 8088



Ejecución de Instrucciones

- Procesar una instrucción consta de dos pasos:
 1. El Procesador **lee** (busca) instrucciones de la memoria, una cada vez.
 2. El Procesador **ejecuta** cada instrucción.
- La ejecución de un programa consiste en repetir el proceso de búsqueda y ejecución de instrucciones.
- Se denomina **ciclo de instrucción** al procesamiento requerido por una única instrucción.



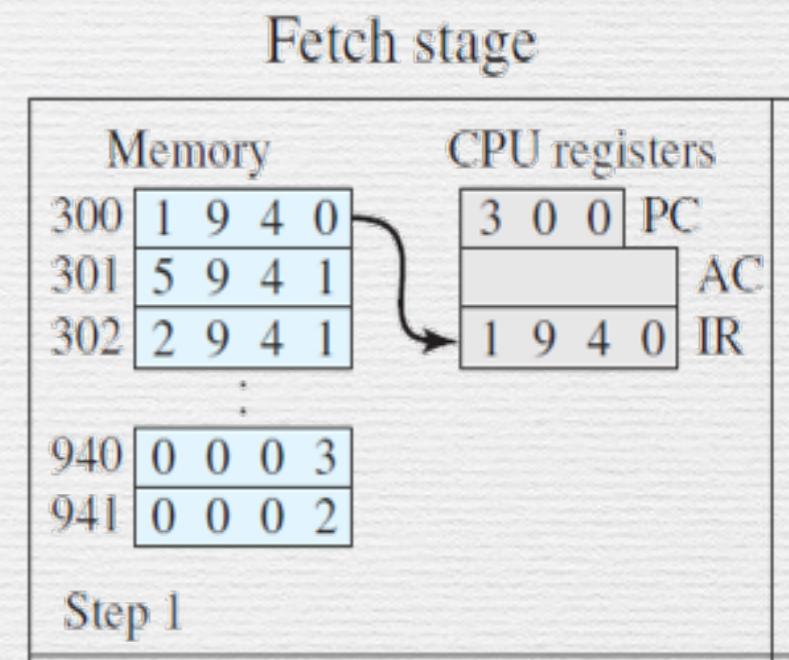
Ejecución de Instrucciones [Beekman06] (pp.128-129)

Definición del problema.

Las instrucciones se codifican con 16 bits:

- 4 para el código de operación
- 12 para el campo de dirección

El programa empieza en la dirección 300 y ocupa 500 posiciones de memoria consecutivas



Step 1 fetch & Step 2 execute

PC = 300, la dirección de la primera instrucción.

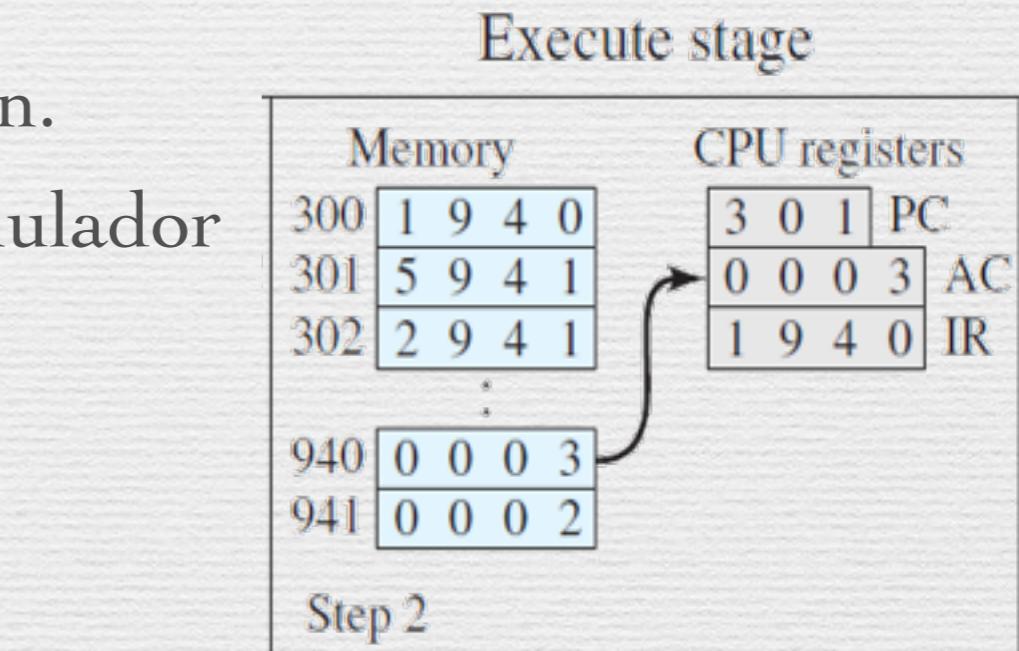
IR = 1 dígito en hexadecimal indica que el acumulador (AC) será cargado desde memoria.

opcode 1 == leer AC

$d(940) = 3$ dígitos en hexadecimal, indican la dirección.

Interpretación: $AC = d(940)$

Ejecución de Instrucciones [Stal05] (pp.14-17)

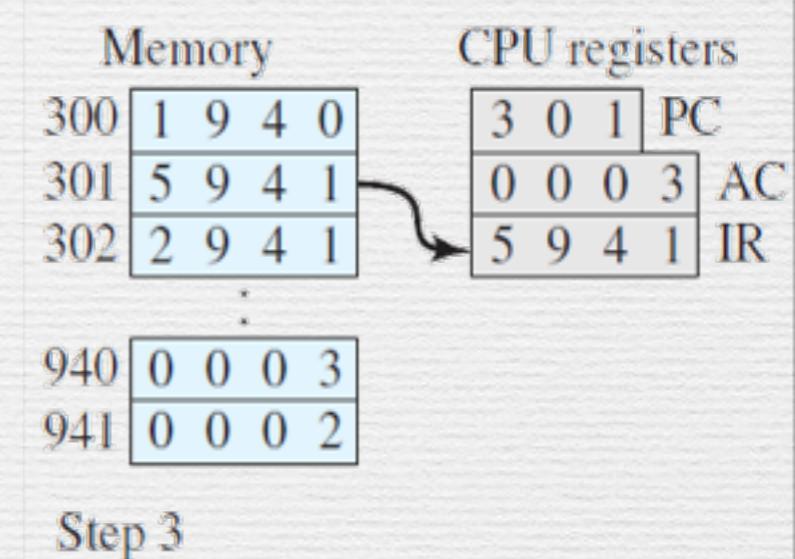


Step 3 fetch & Step 4 execute

El PC se incrementa.

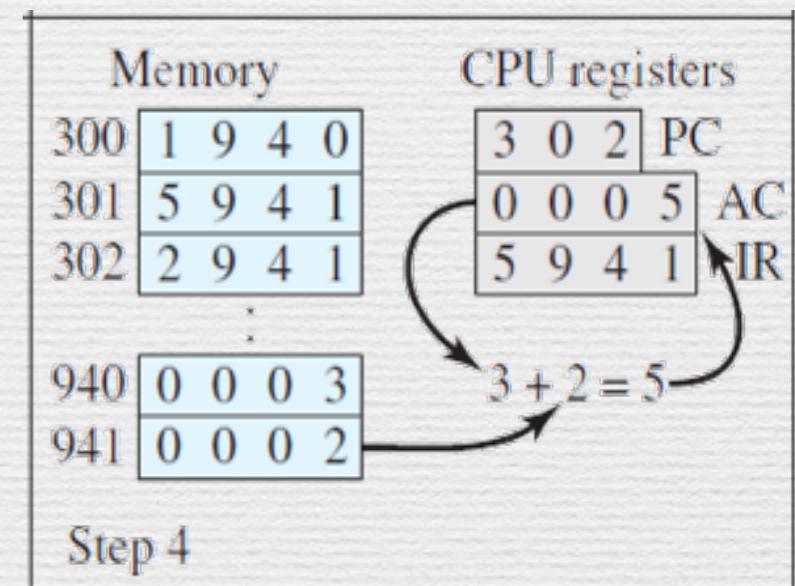
IR = instrucción (5941) desde la dirección 301.

opcode 5 == sumar



Interpretación: El anterior contenido del AC y el contenido de la dirección 941 se suman y el resultado se almacena en el AC.

$$AC = AC + d(941)$$



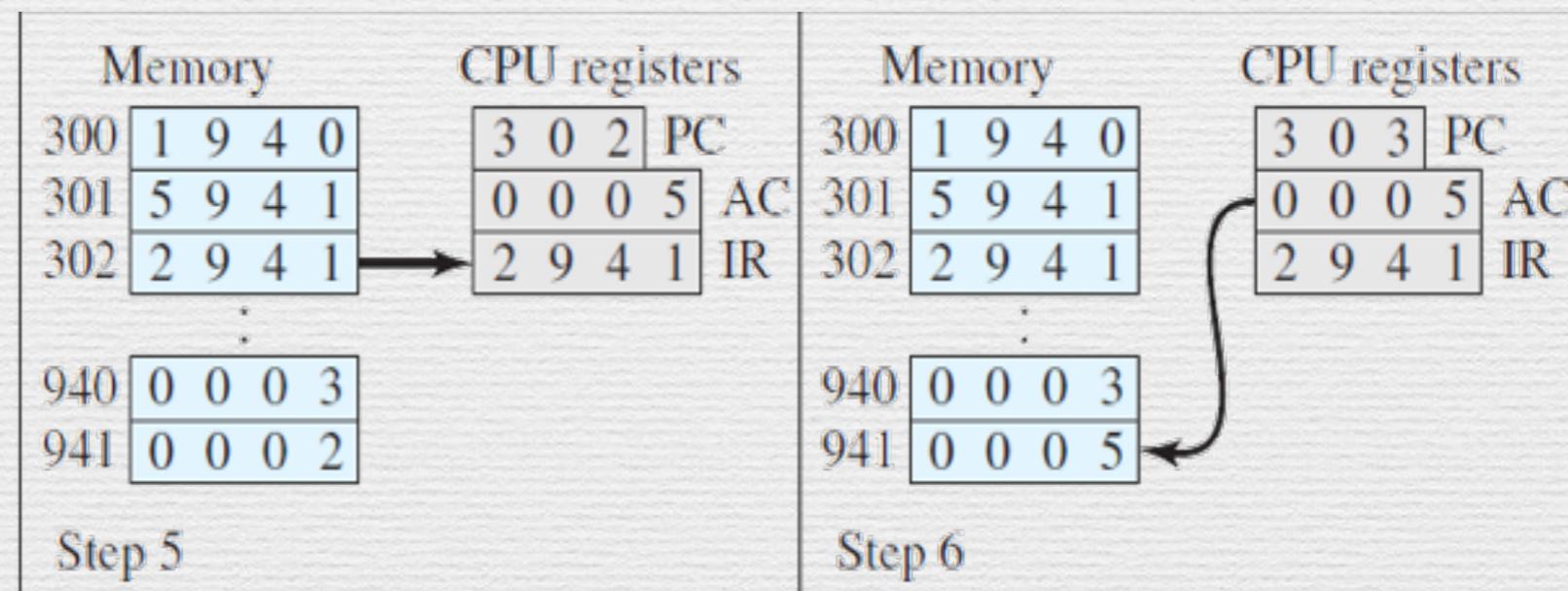
Ejecución de Instrucciones [Stal05] (pp.14-17)

[Stal12] (pp.31-33)

Step 5 fetch & Step 6 execute

1. La siguiente instrucción (2941) será captada desde la dirección 302. El PC se incrementa.
2. El contenido del AC se aloja en la dirección 941, según indica el opcode 2.

Interpretación: $d(941) = AC$



Ejecución de Instrucciones [Stal05] (pp.14-17)

[Stal12] (pp.31-33)

Podemos clasificarlas en:

- Transferencia de datos

 MOVER ORIGEN, DESTINO

- Aritmético-lógicas

 SUMAR op1, op2, resultado

 COMPARAR Ro,R1

- Entradas/salidas

 IN puerto, destino

 OUT origen, puerto

- Control:

 JUMP/JNE/JE etiqueta

 CALL funcion

 RETURN e IRETURN

 HALT

Ejemplo: sumar dos número cargados en memoria en las direcciones direc1 y direc2, y poner el resultado en direc3.

MOVER direc1, Ro

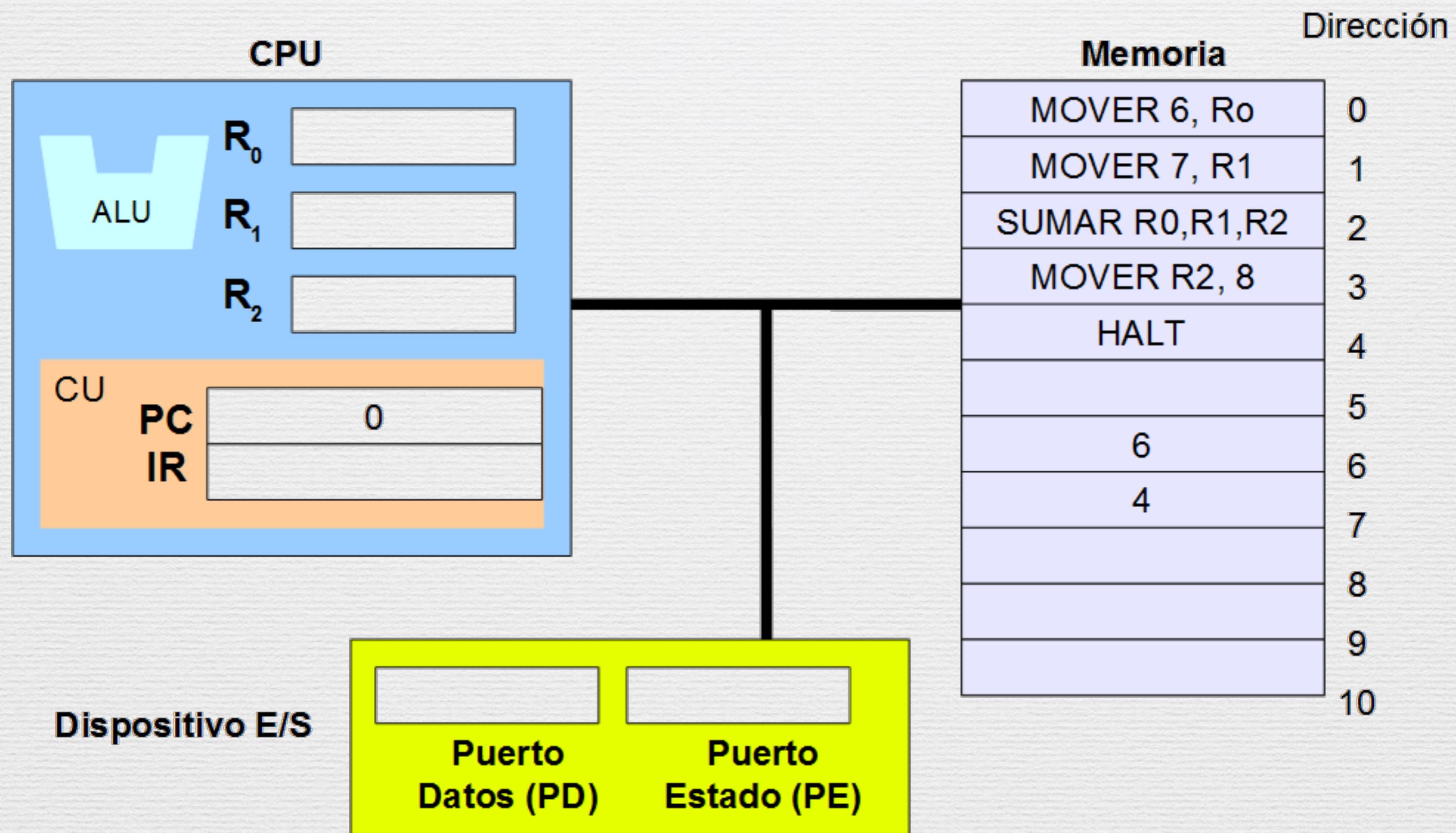
MOVER direc2, R1

SUMAR Ro,R1,R2

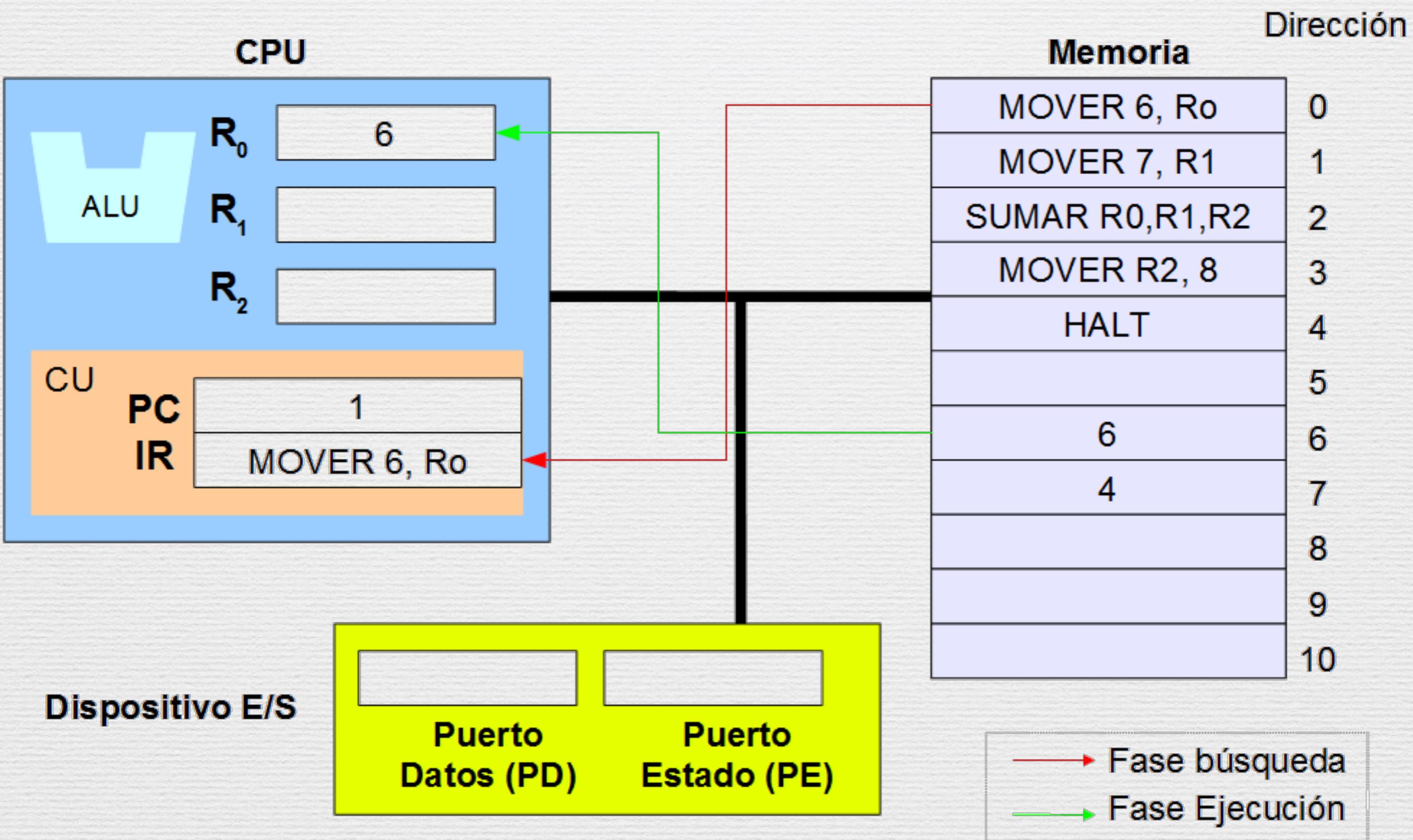
MOVER R2, direc3

HALT

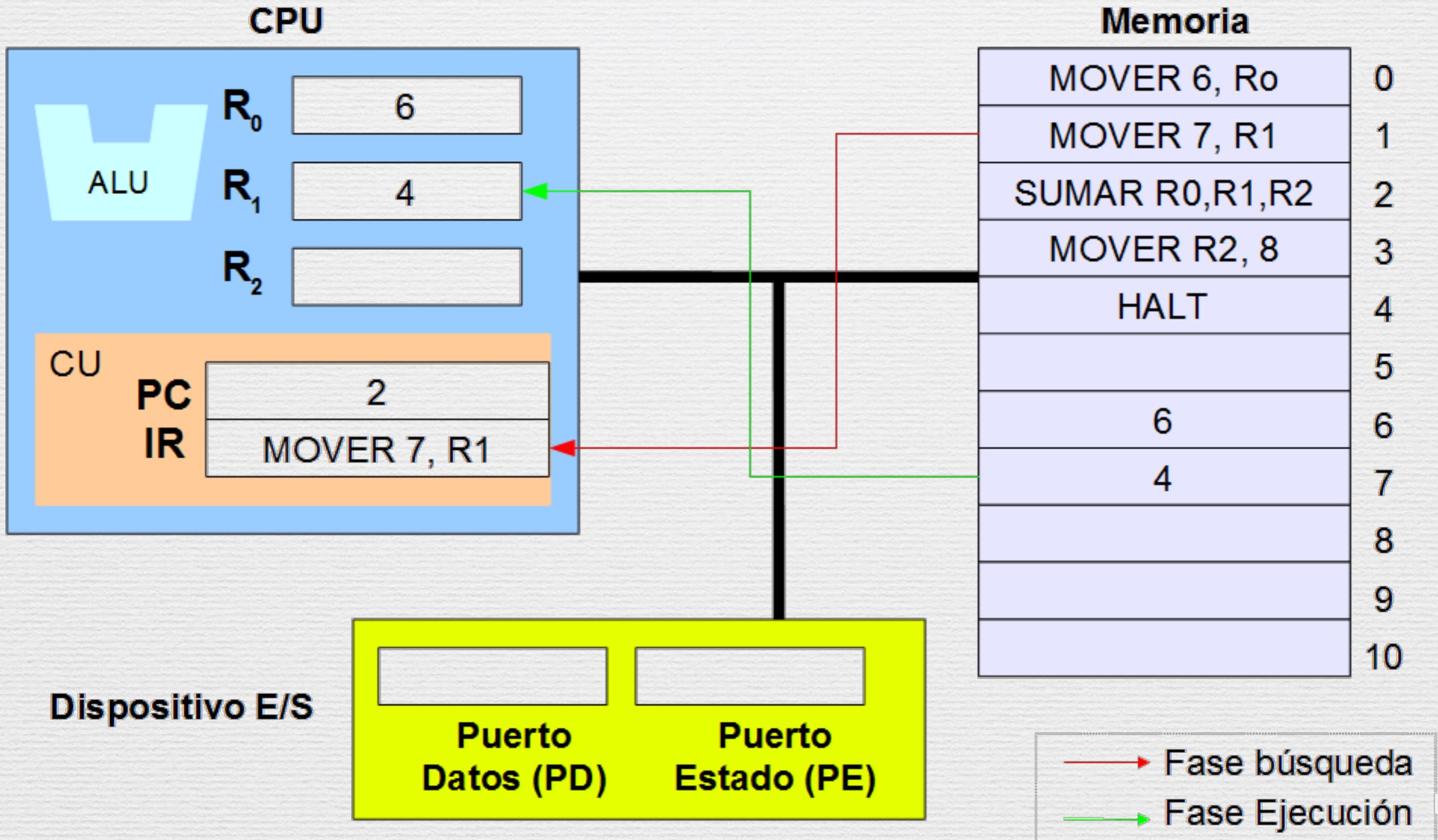
Ejecución del ejemplo (paso 0)



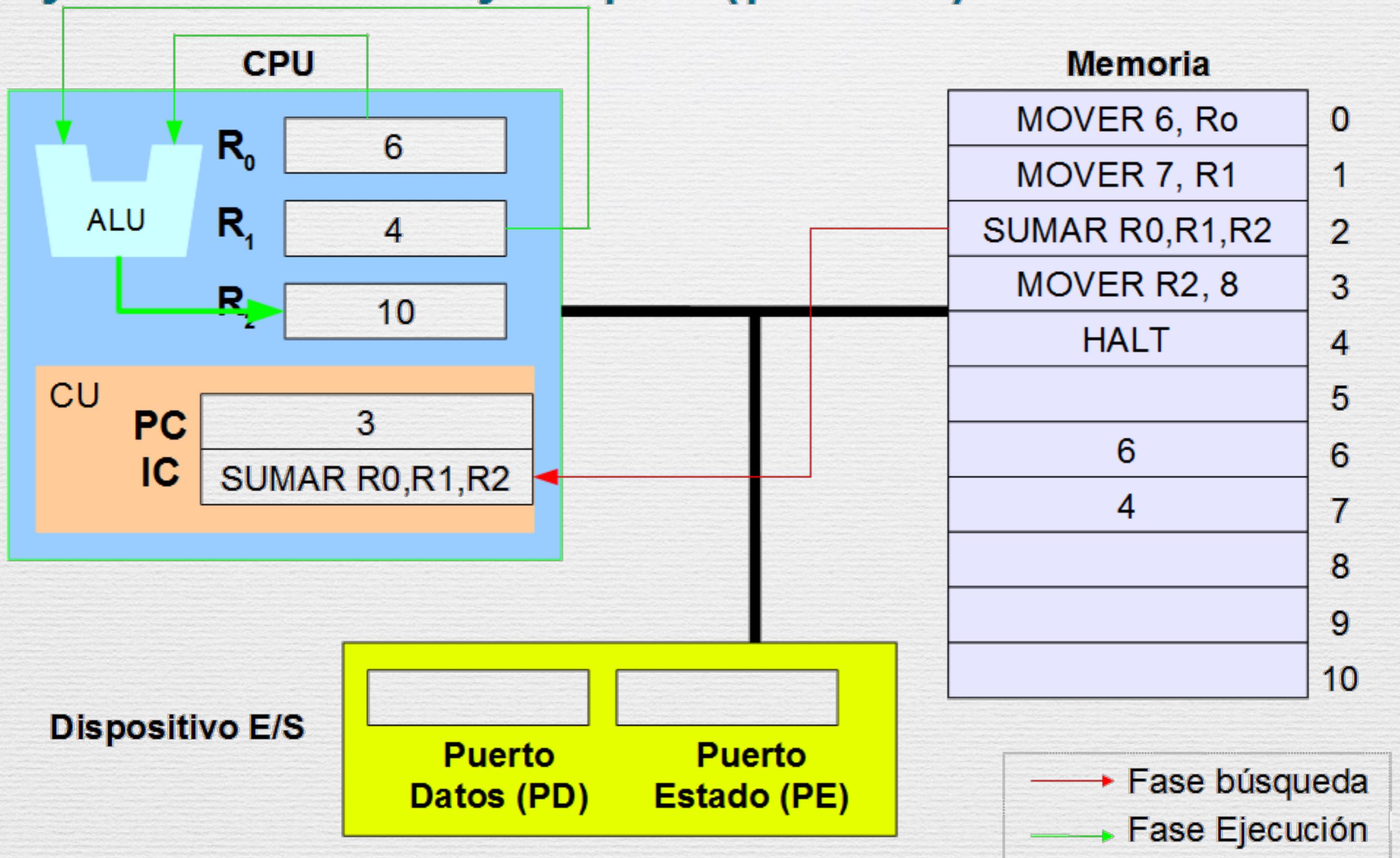
Ejecución del ejemplo (paso 1)



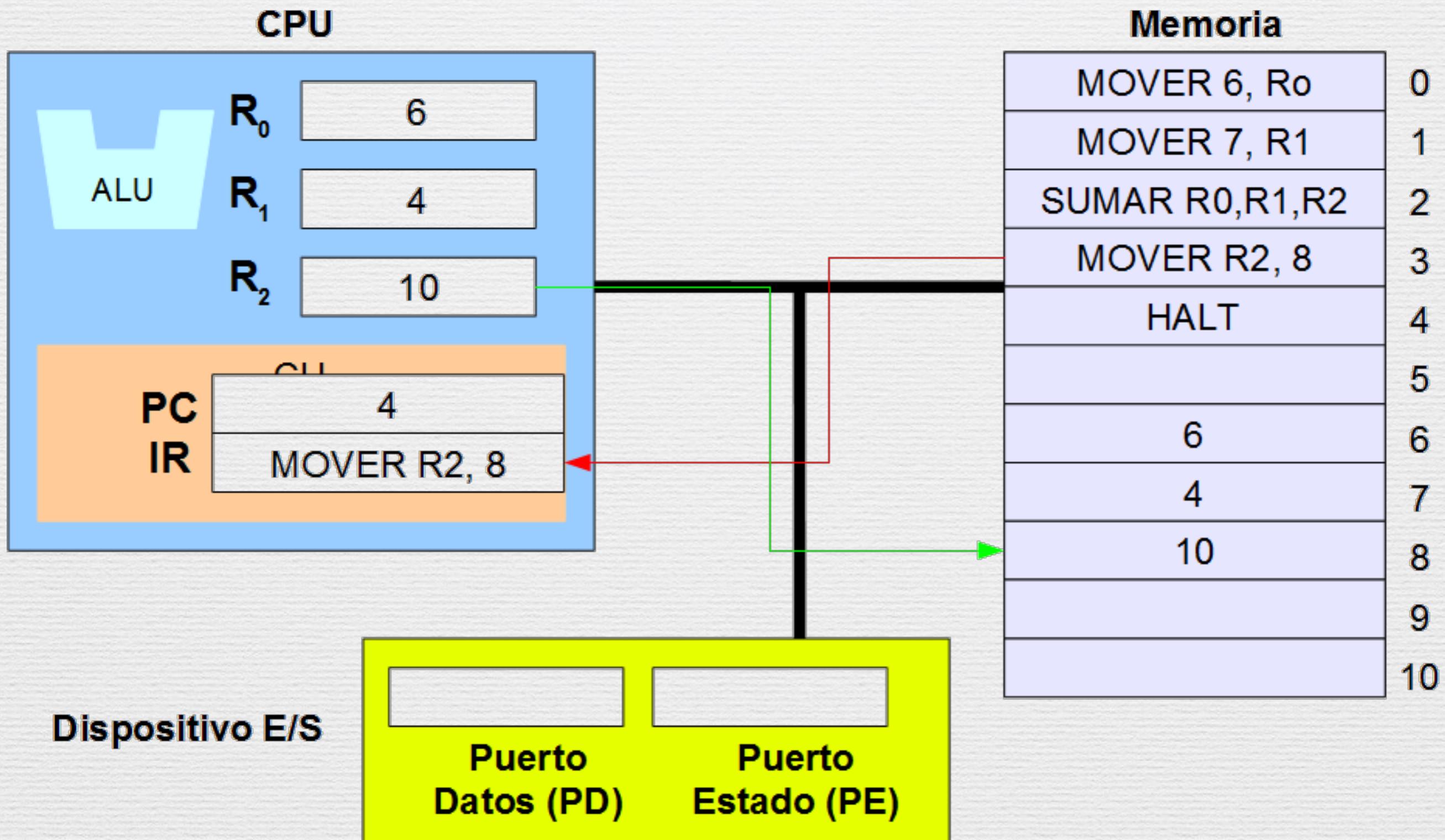
Ejecución del ejemplo (paso 2)



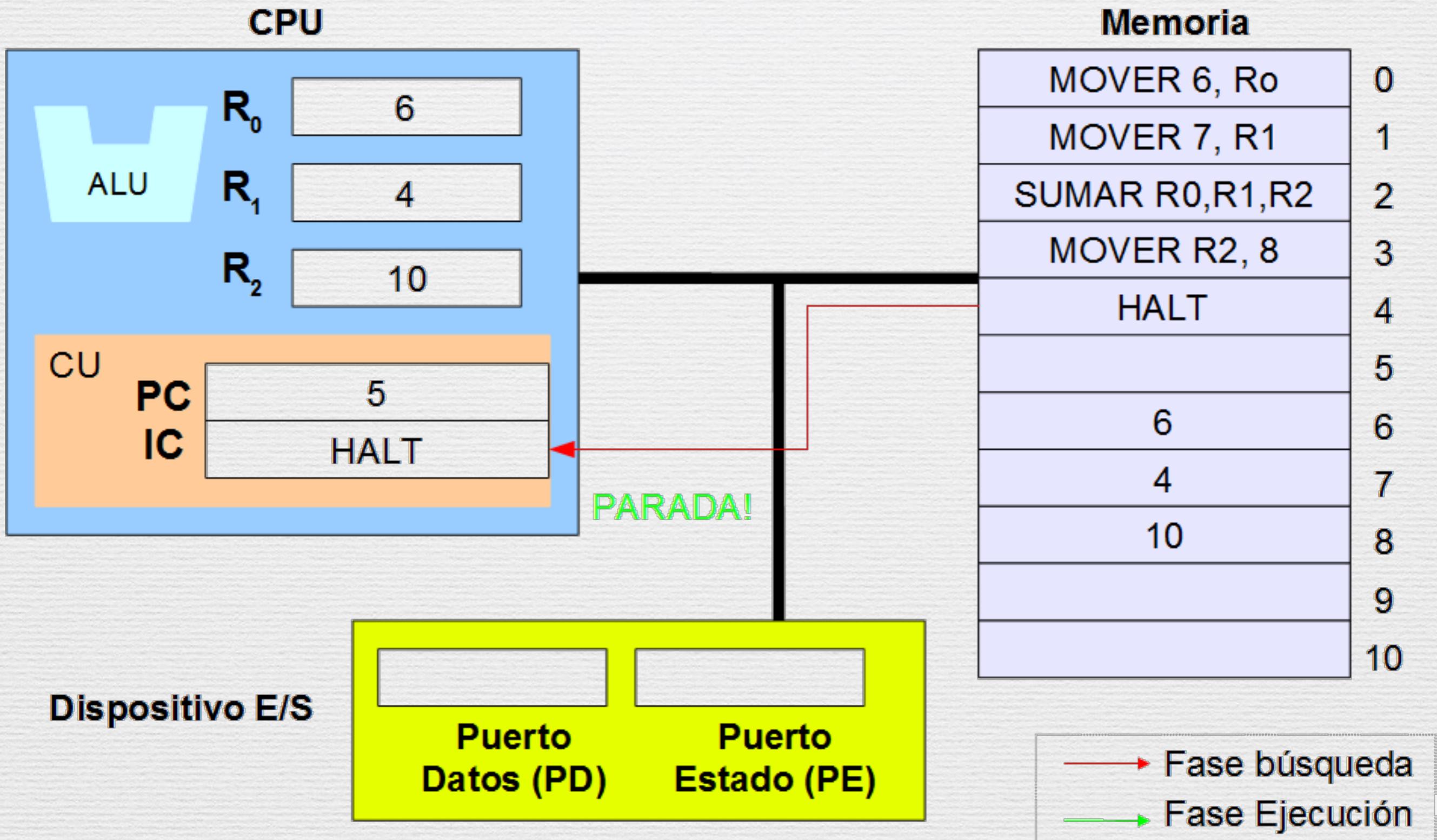
Ejecución del ejemplo (paso 3)



Ejecución del ejemplo (paso 4)



Ejecución del ejemplo (paso 5)



Técnicas de Comunicaciones E/S

¿Qué sucede cuando un programa desea comunicarse con algún dispositivo de E/S? ojo!! los dispositivos E/S son muy lentos

Hay tres técnicas para llevar a cabo las operaciones de E/S:

- E/S Programada -> programmed IO
- E/S Dirigida por Interrupciones -> Interrupt-driven IO
- Acceso Directo a Memoria -> Direct Access Memory / DMA

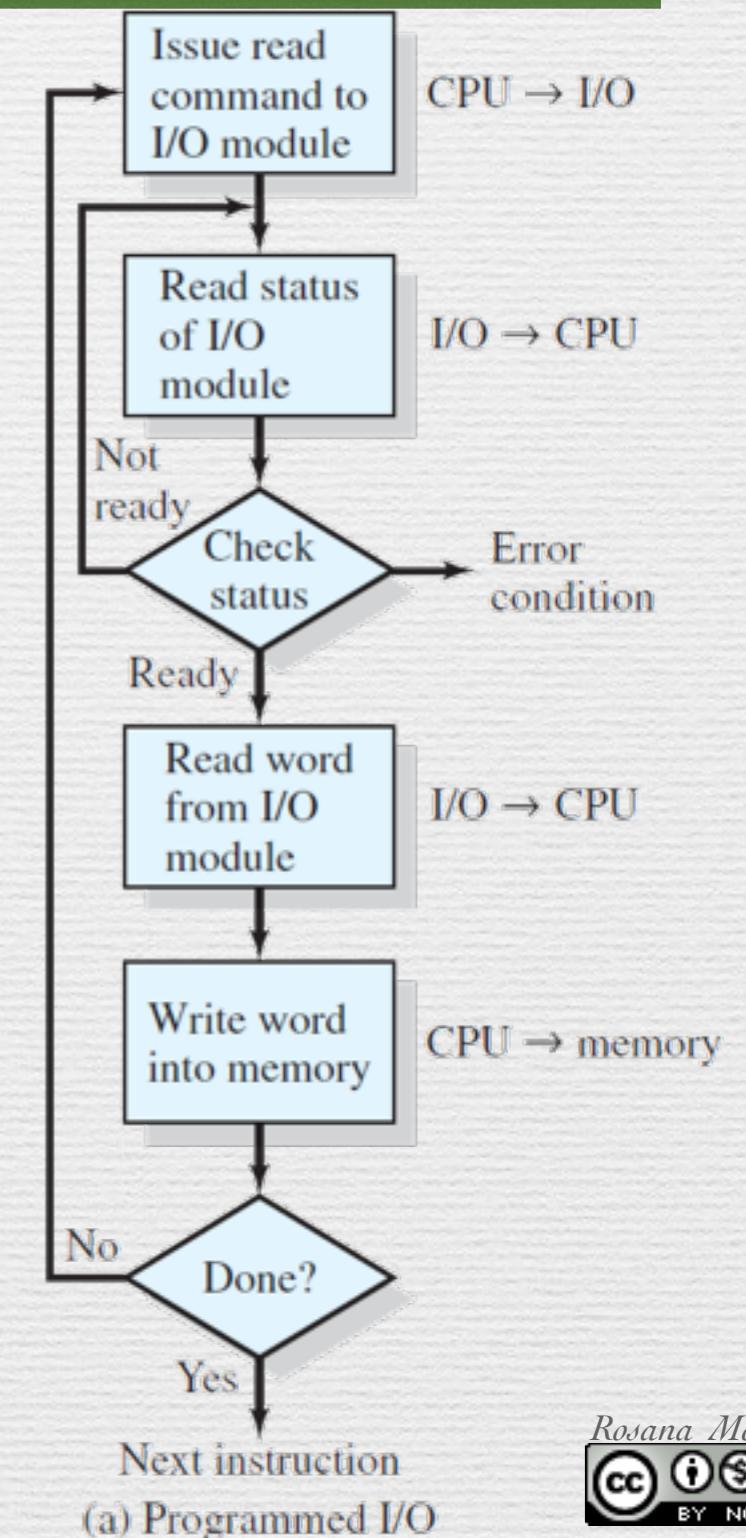
Interrupciones [Stal05] (pp.34-37) [Stal12] (pp.34-43)

E/S Programada

- El procesador encuentra una instrucción con la E/S. Se genera un mandato al módulo de E/S apropiado.
- El procesador adopta un papel activo mientras se atiende la instrucción de E/S y comprueba periódicamente el estado de la ejecución del módulo de E/S hasta que comprueba que ha finalizado la operación.

Problema: El procesador pasa mucho tiempo esperando la finalización del módulo de E/S y el sistema se degrada gravemente.

Solución: Mientras se atiende al módulo de E/S, el procesador pueda continuar con trabajo útil.

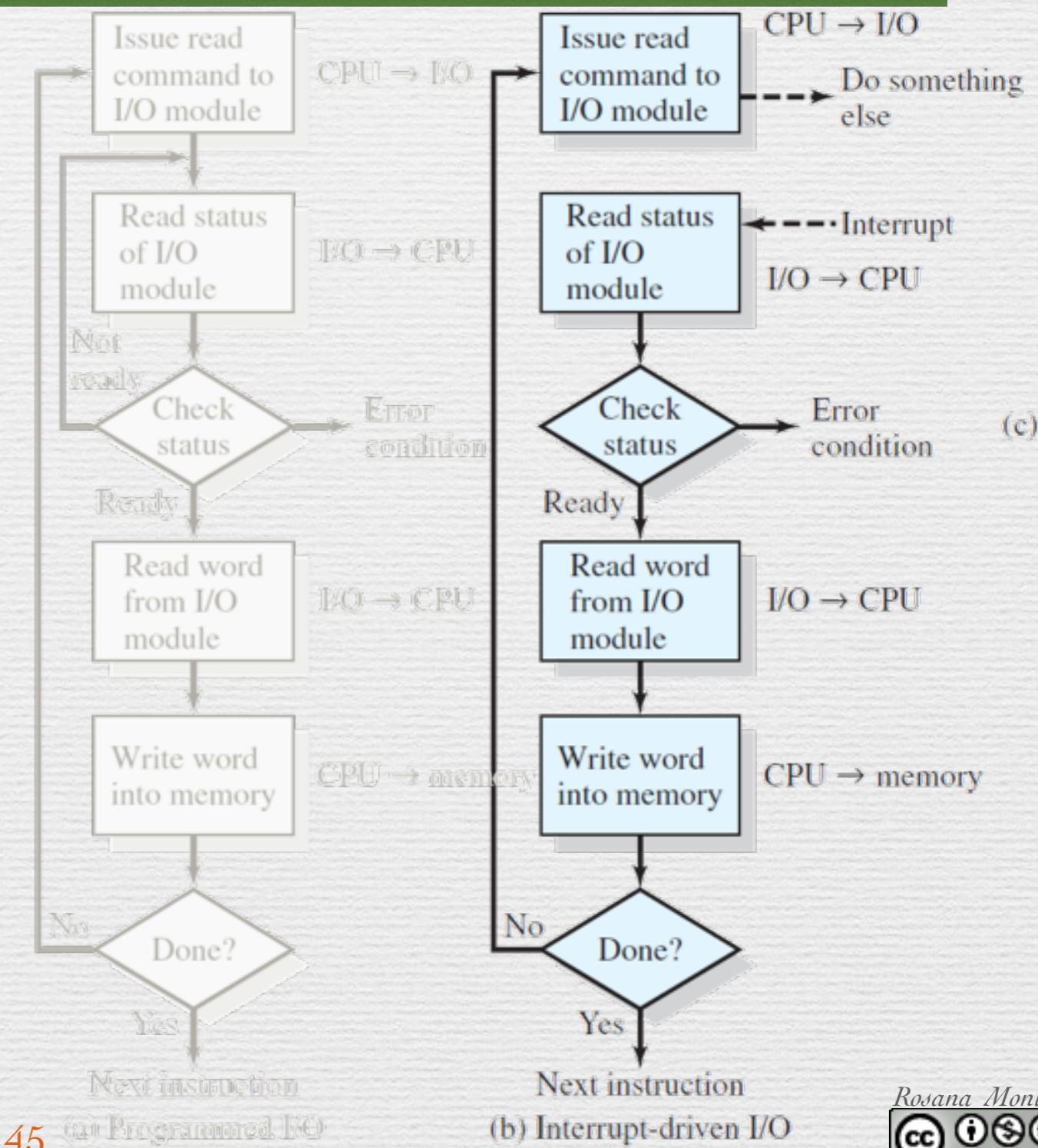


E/S dirigida por interrupciones

- Evento que interrumpe el flujo normal de ejecución producido por un elemento externo al procesador. Es un evento asíncrono.

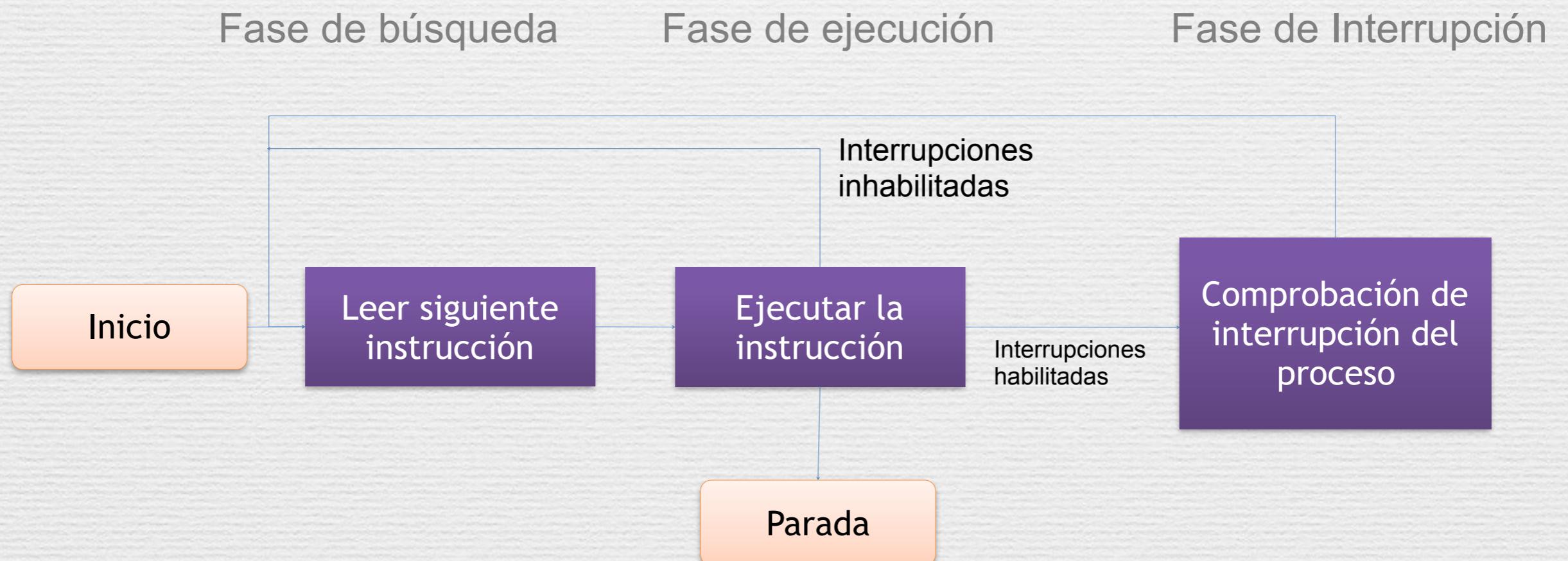
Problema: En transferencias considerables de memoria a dispositivo o viceversa conlleva un uso excesivo del procesador.

Solución: Acceso Directo a Memoria. En un solo mandato se genera todo lo necesario para realizar la transferencia de información de memoria al dispositivo o viceversa.

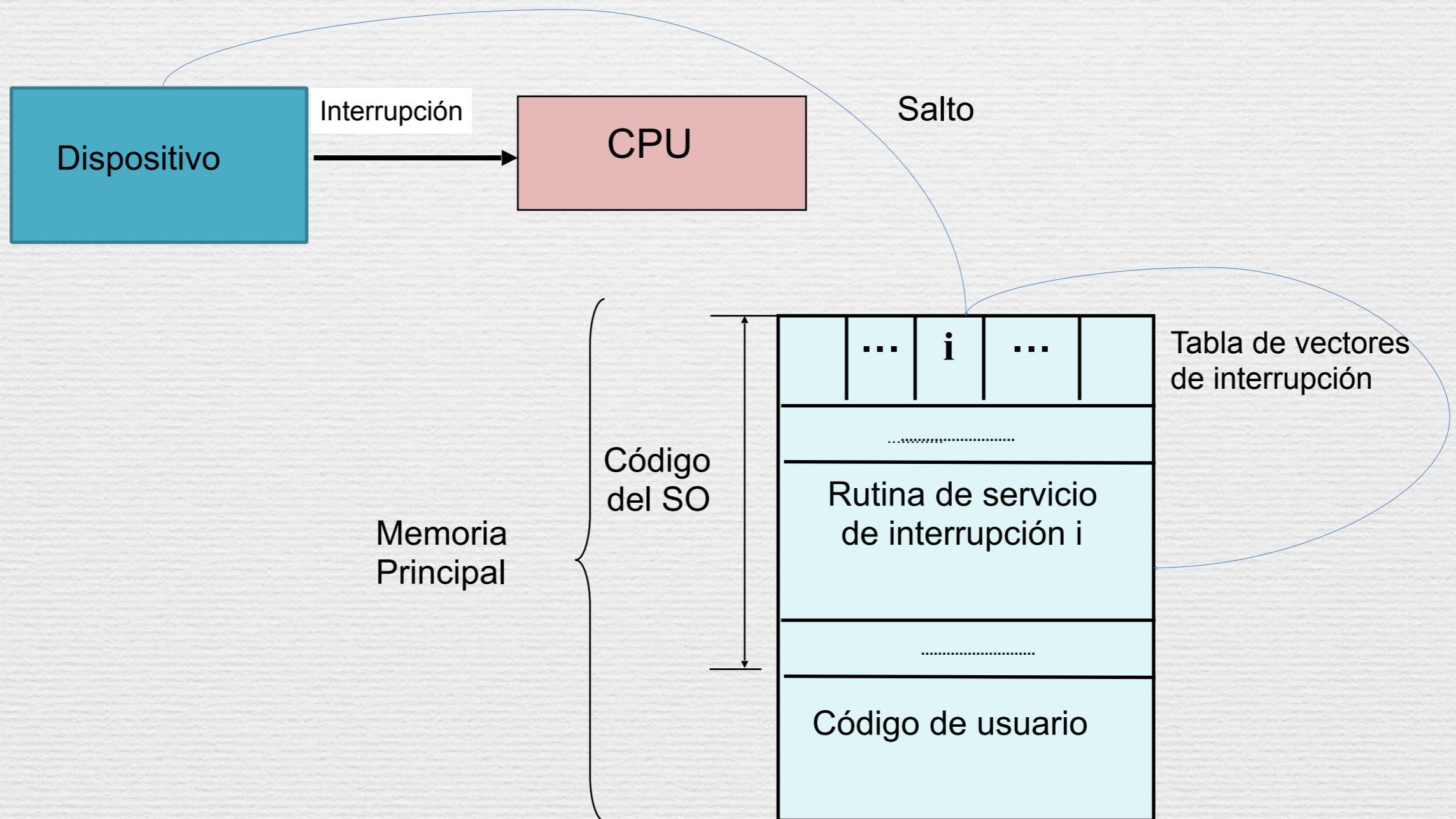


Ciclo de instrucción

- Ciclo de instrucción con interrupciones.



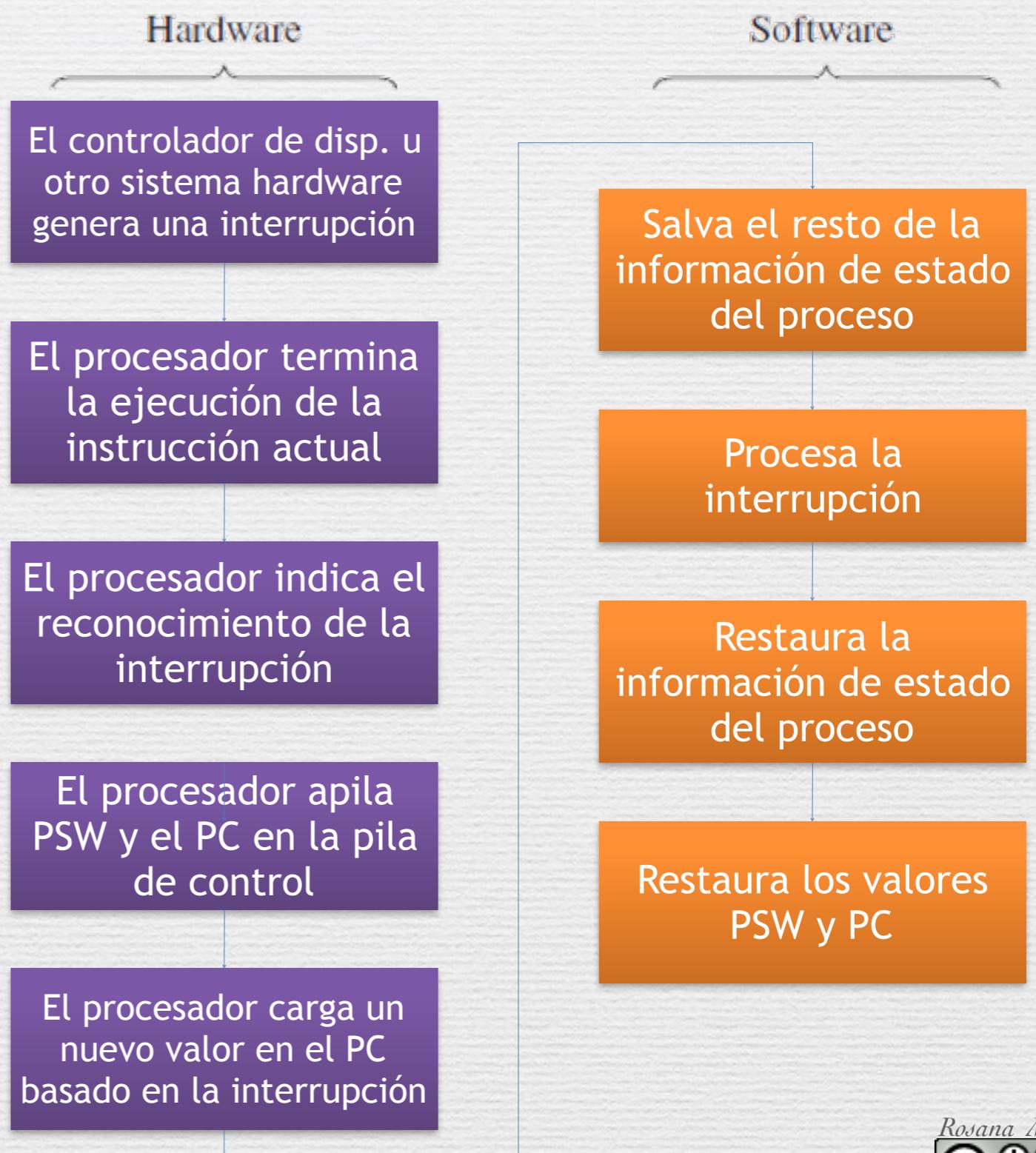
Interrupciones Vectorizadas



1.2 Capa Hardware

Tratamiento de Interrupciones Vectorizadas

[Stal05] (pp. 23. Fig. 1.10)
[Stal12] (pp. 39. Fig. 1.10)

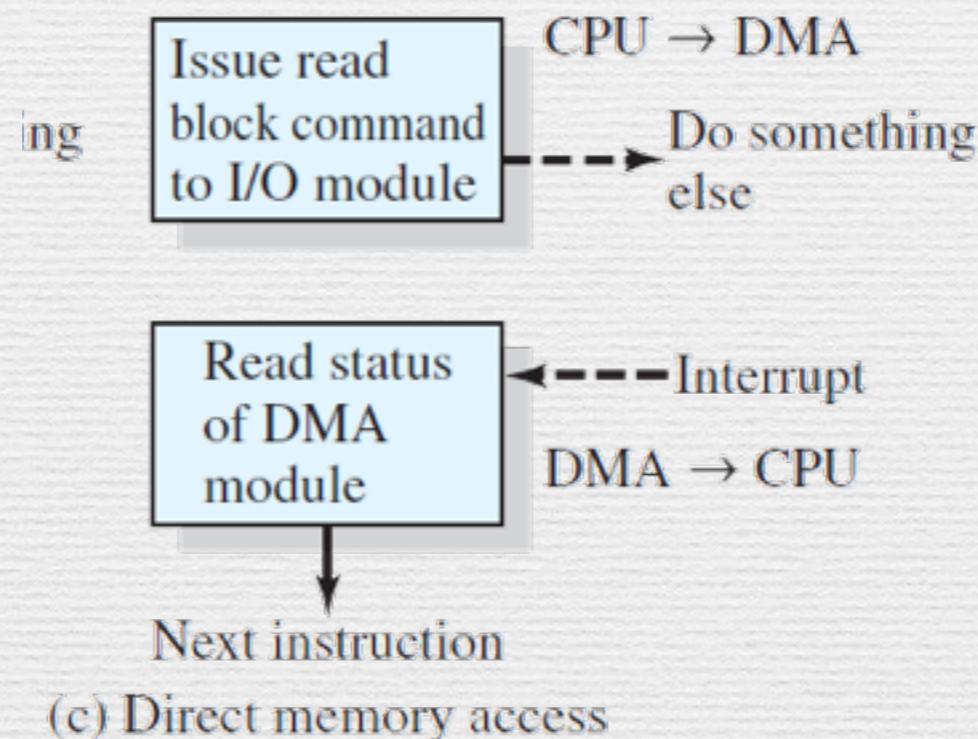


1.2 Capa Hardware

Técnicas de Comunicación de E/S [Stal05] (pp.34-37)

[Stal12] (pp.34-43)

- **Acceso Directo a Memoria (DMA, Direct Access Memory).** Realizada por un módulo separado conectado en el bus del sistema o incluida en un módulo de E/S. Útil cuando el procesador desea leer o escribir un bloque de datos.



Protección Hardware

Protección de los Dispositivos de E/S [Carr07] (pp.25-28)

- Los dispositivos de E/S son recursos que han de estar protegidos (ejemplo, los archivos, las impresoras, ...)
- ¿Cómo se consigue? → Las instrucciones máquina para acceso a los dispositivos de E/S no pueden ejecutarse en modo usuario: son privilegiadas.
- Cualquier acceso a los dispositivos desde un programa de usuario se hará mediante peticiones al SO.

Protección de Memoria

- Cada programa en ejecución requiere de un espacio de memoria.
- **Objetivo:** Hay que proteger la zona de memoria asignada y la memoria en la que está el código del sistema operativo (tabla de vectores de interrupción, rutinas de tratamiento de cada interrupción).

Protección Hardware

Protección [Carr07] (pp.4)

Funcionamiento en Modo Dual. ¿Qué ocurre si un programa accede a la memoria donde se alojan los vectores de interrupciones? ¿Qué pasa si las modifica?

Solución: El procesador dispone de diferentes modos de ejecución de instrucciones:

- **Instrucciones privilegiadas (modo supervisor/kernel):** Aquellas cuya ejecución puede interferir en la ejecución de un programa cualquiera o programa del SO (ejemplo, escribir en el puerto de un dispositivo).
- **Instrucciones no privilegiadas (modo usuario):** Aquellas cuya ejecución no presenta ningún problema de seguridad para el resto de programas (ejemplo, incrementar un contador).

Software

- ♦ Es la parte intangible del ordenador
- ♦ Tipos de software:
 - ♦ Software de Sistemas
 - ♦ Software de Aplicación
 - ♦ Software estándar
 - ♦ Procesadores de texto
 - ♦ Hojas de Cálculo
 - ♦ Bases de Datos
 - ♦ Software de presentaciones
 - ♦ Software a medida

Aplicaciones integradas y suites [Beekman06] (pp.133-134)

Software

- ♦ Es la parte intangible del ordenador
- ♦ Tipos de software:
 - ♦ Software de Sistemas
 - ♦ Sistemas Operativos
 - ♦ Lenguajes de programación
 - ♦ Utilidades del Sistema
 - ♦ Software de Aplicación

Software de sistema [Beekman06] (pp.135-140)

Clasificación Software

El Software son los programas de aplicación y los sistemas operativos, que según las funciones que realizan pueden ser clasificados en:

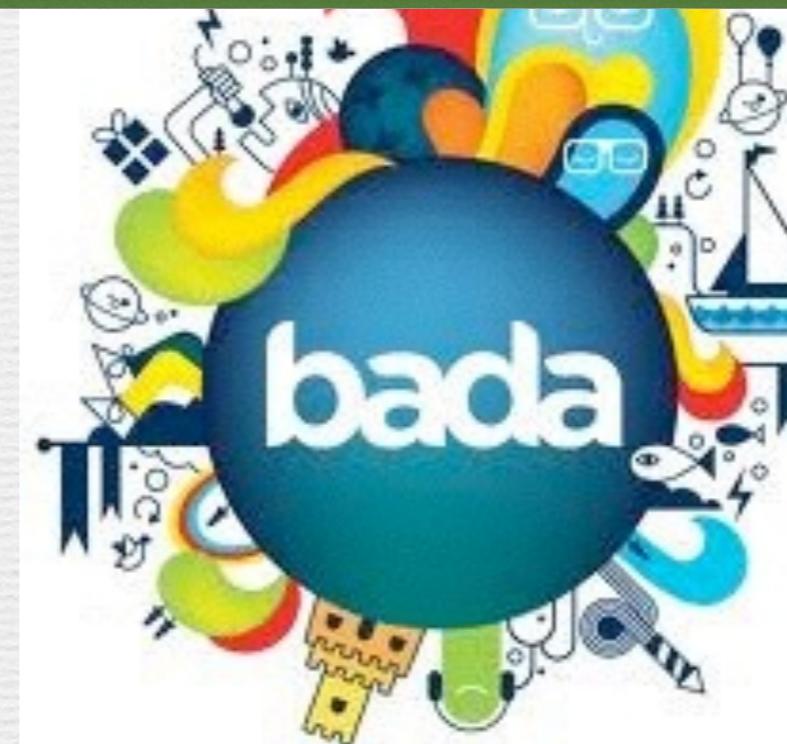
- Software de sistema.
- Software de aplicaciones.
- Software de utilidad

Software de sistema

- Conjunto de programas que sirven para interactuar con el sistema, confiriendo control sobre el hardware, además de dar soporte a otros programas.



Plataformas



Plataformas hardware y software [Beeckman06] (pp.145-147)



Software de aplicación

- Programas diseñados para los usuarios para facilitar la realización de tareas específicas en el ordenador, como pueden ser las aplicaciones ofimáticas u otros tipos de software especializado.
- Software a medida.



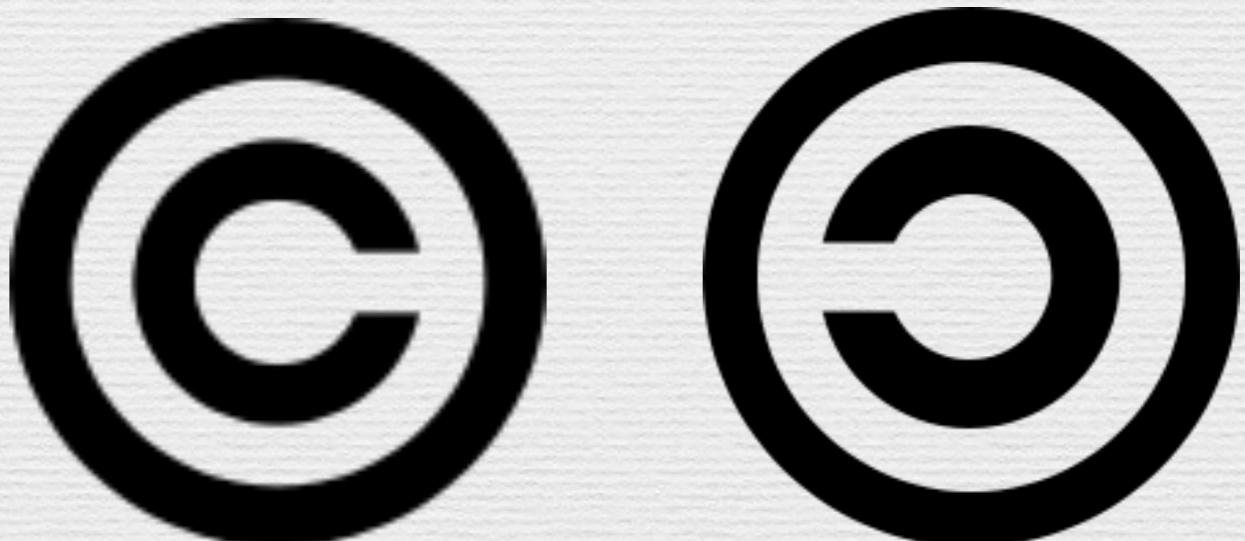
Software de utilidad

- Programas especiales que sirven para revisar ciertos aspectos del funcionamiento del sistema, tanto de hardware como de software.



Clasificación del Software

- Se desea clasificar el software atendiendo a su propietario.
- Implica el conocimiento de las licencias para contenidos digitales. Ej.: copyright, copyleft.
- Implica el conocimiento de la Ley de Propiedad Intelectual.



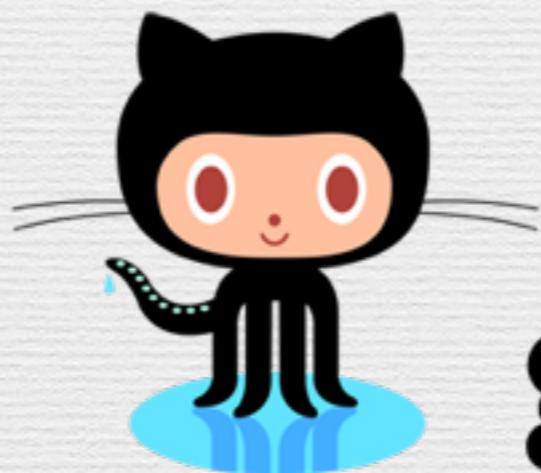
Software propietario =

- Se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible, o que el acceso a éste se encuentra restringido.



Software libre =

- El software libre es aquel que puede ser distribuido, modificado, copiado y usado.
- Debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan.



github
SOCIAL CODING



Libertades del Software Libre

- **Libertad 0**
propósito.
- **Libertad 1**
y adaptarlo a tus necesidades.
- **Libertad 2**
ayudar a los demás.
- **Libertad 3**
las mejoras, de modo que todos se beneficien.

Clasificación del Software según licencia de uso

- ♦ Comercial modificable.
- ♦ Shareware
- ♦ Freeware
- ♦ Open source varios tipos de licencia.
- ♦ Otros términos conocidos:
<http://elidiomadelaweb.com/malware/20-terminos-acabados-en-ware/>

Licencias

“Una licencia de software es un contrato entre el licenciante y el licenciatario del programa informático, para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas”.

Centro de Software Libre (2014)

Licencias

- EULA (End User License Agreement): licencia por la cual el uso de un producto sólo está permitido para un único usuario final (el comprador).
- GNU GPL, (Licencia Pública General) es una licencia creada por la FSF (Free Software Foundation) a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Uso de la propiedad intelectual



Reconocimiento (by)

- Obliga a citar al autor



No comercial (nc)

- Impide usar los contenidos para un beneficio económico



Sin obras derivadas (nd)

- Obliga a que esa obra sea distribuida inalterada

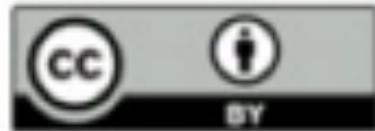


ShareAlike (sa)

- Obliga a que las obras derivadas se distribuyan siempre bajo la misma licencia del trabajo original (copyleft)

Ver también: Licencias de uso de la propiedad intelectual en
<https://creativecommons.org/licenses/>

Licencias CC: ejemplos



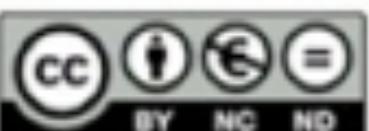
Reconocimiento (BY): se permite cualquier explotación de la obra, incluyendo la comercial, así como la creación de obras derivadas, la distribución de las cuales también está permitida sin ninguna restricción.



Reconocimiento-No comercial (BY-NC): no se permite el uso comercial de la obra original ni de las posibles obras derivadas.



Reconocimiento-No comercial-Compartir Igual (BY-NC-SA): no se permite el uso comercial de la obra original ni de las posibles obras derivadas, las cuales deberán mantener el mismo tipo de licencia.



Reconocimiento-No comercial-Sin obra derivada (BY-NC-ND): no se permite el uso comercial de la obra original ni la generación de obras derivadas.

Compatibilidad

Creative Commons

Tabla de compatibilidad		Términos que pueden usarse en un trabajo derivado						
		BY	BY-NC	BY-NC-ND	BY-NC-SA	BY-ND	BY-SA	PD
Trabajo original	PD	✓	✓	✓	✓	✓	✓	✓
	BY	✓	✓	✓	✓	✓	✓	✗
	BY-NC	✗	✓	✓	✓	✗	✗	✗
	BY-NC-ND	✗	✗	✗	✗	✗	✗	✗
	BY-NC-SA	✗	✗	✗	✓	✗	✗	✗
	BY-ND	✗	✗	✗	✗	✗	✗	✗
	BY-SA	✗	✗	✗	✗	✗	✓	✗

fuente: Creative Commons, CC BY http://wiki.creativecommons.org/Frequently_Asked_Questions

1.3 El Sistema Operativo

El Sistema Operativo [Stal05] (cap.2, pp.53-79) [Stal12] (pp.68-82,91-96)

Un SO es un programa o conjunto de programas que controla la ejecución de los programas de aplicación y que actúa como interfaz entre el usuario de una computadora y el hardware de la misma.

Utilidades y Herramientas

Sistema Operativo

Hardware

1.3 El Sistema Operativo

Características deseables

- **Comodidad en el uso.** (Convenience)
- **Eficiencia:** Existen más programas que recursos. Hay que repartir los recursos entre los programas
- **Facilidad de Evolución:** Un SO importante debe evolucionar en el tiempo por las siguientes razones:
 - Actualizaciones del hardware y nuevos tipos de hardware.
 - Mejorar y/o aportar nuevos servicios.
 - Resolución de fallos.

The flipped  classroom

1.4 Utilidades del Sistema

Programas de Servicio del SO [Prie06] (Cap.13, sección 13.1, pp.518-520)

Se trata de un conjunto de programas de servicio que, en cierta medida, pueden considerarse como una ampliación del SO:

- Compactación de discos.
- Compresión de datos.
- Gestión de comunicaciones.
- Navegadores de internet.
- Respaldo de seguridad.
- Recuperación de archivos eliminados.
- Antivirus.
- Salvapantallas.
- Interfaz gráfica o Graphics User Interface (GUI).



Entornos GNU/Linux

- GNU/Linux y todas las variantes Unix no fueron creados con una interfaz visual de usuario.
- Representa una forma de interacción con el sistema a través de líneas de texto secuenciales (líneas de comando).
- Casi la totalidad de las aplicaciones pueden ser configuradas desde la consola o terminal.

Abstracciones del SO

Permiten interacciones más sencillas con el usuario:

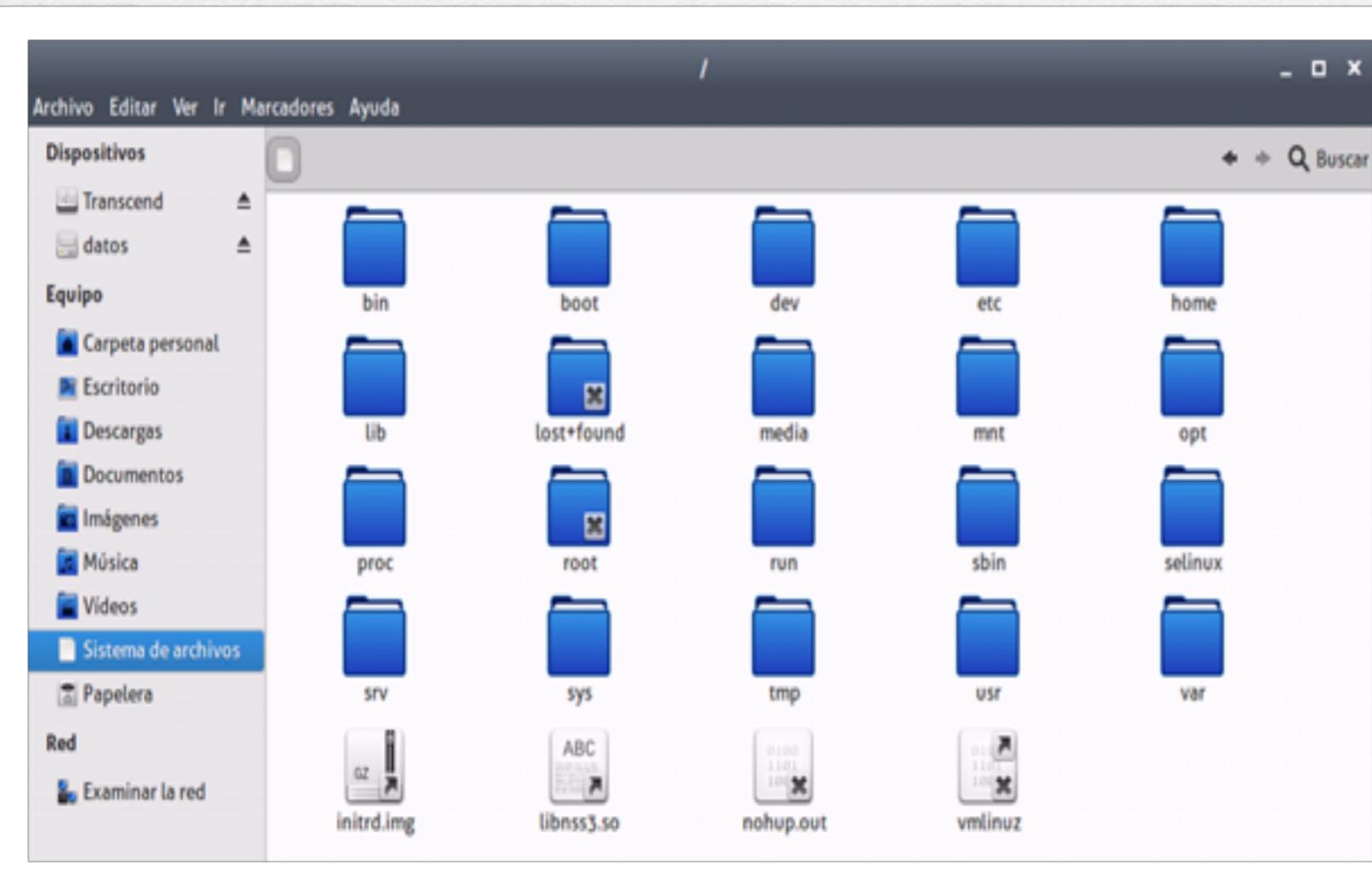
- Concepto de archivo/fichero
- Concepto de carpeta/directorio
- Concepto de unidad

Aplicaciones

Descargas

Escritorio

Imágenes



1.4 Utilidades del Sistema

Herramientas Generales

Su misión es facilitar la construcción de las aplicaciones de los usuarios, sea cual sea la naturaleza de éstas, tales como:

- Editores de texto.
- Compiladores.
- Intérpretes.
- Enlazadores.
- Cargadores/Montadores.
- ...

A nivel de programador

A word cloud illustrating various programming frameworks and languages, with the size of the text indicating its relative popularity or prevalence. The most prominent words are 'Ruby', 'Python', 'PHP', 'Java', 'Rails', 'Struts', 'Django', and 'JSF'. Other visible words include 'Spring', 'Tapestry', 'Symphony', 'Turbogears', 'Rails', 'Grails', 'NET', 'Smalltalk', 'Cocoon', and 'JSF'.

Ruby Tapestry
Programming symphony Python
Turbogears develop Web Framework learn .NET
Rails build Arc Lisp Struts Seaside Smalltalk
java Django
PHP Cocoon JSF

A nivel de programador



(18)

Autoevaluación

Entre teoría y prácticas yo puedo:

- ♦ Explicar la relación entre hardware y software
- ♦ Conocer cómo se representa la información que manipula un ordenador.
- ♦ Entender el concepto de programa y los engranajes que activan ejecución de un conjunto de instrucciones.
- ♦ Describir las tres categorías fundamentales de software y sus relaciones.
- ♦ Esbozar la evolución de las interfaces de usuario.
- ♦ Analizar qué aplicación es necesaria en las distintas disciplinas de un profesional digital.

Y ademas ...