

## Contenidos

### 1.1 Componentes de un Sistema de Cómputo.

- 1.1.1 Definiciones Básicas.
- 1.1.2 Registros del Procesador.
- 1.1.3 Ejecución de Instrucciones. Tipos de Instrucciones.

### 1.2 Capa Hardware.

- 1.2.1 Estructura de un Ordenador.
- 1.2.2 Técnicas de Comunicación de E/S.

### 1.3 El Sistema Operativo.

### 1.4 Utilidades del Sistema.

## Objetivos

- Conocer los elementos principales de un Sistema de Cómputo.
- Disponer los elementos de la parte hardware.
- Conocer el software más próximo a la capa hardware: el Sistema Operativo.
- Conocer las principales utilidades software que se utilizan en un sistema de cómputo.

## Bibliografía básica

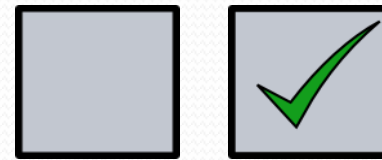
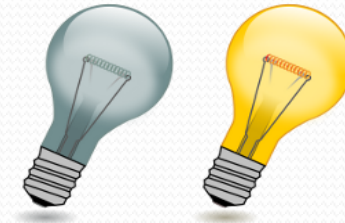
- [Prie06] A. Prieto, A. Lloris, J.C. Torres, **Introducción a la Informática**, McGraw-Hill, 2006
- [Stal05] W. Stallings, **Sistemas Operativos, Aspectos Internos y Principios de Diseño (5ª Edición)**. Pearson Education, 2005
- [Carr07] J. Carretero, F. García, P. de Miguel, F. Pérez, **Sistemas Operativos (2ª Edición)**, McGraw-Hill, 2007

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

- Informática, Computador, Bit, ...
- Instrucción u Orden, Programa, Lenguaje de Programación.
- Lenguaje Máquina.
- Hardware (soporte físico) y Firmware.
- Software (soporte lógico).
- Sistema Informático.
- ...

## Definición de Bit

- Bit (**B**inari **D**igit): unidad mínima de información
- Codifica información:
  - 1 bit: 0 ó 1
  - 2 bits: 00, 01, 10 ó 11
  - ...
- $2^{n^0}$  de bits = elementos
- $\text{Log}_2 \text{ elementos} = n^0 \text{ de bits}$



0

1

False

True

## Múltiplos del bit y Unidades de medida

- 1 Byte (B) = 8 bits (b) (Byte: Unidad mínima para direccionar)

Bytes	bits
1 Kilobyte (KB) = $2^{10}$ B	1 Kilobit (Kb) = $2^{10}$ bits
1 Megabyte (KB) = $2^{10}$ KB	1 Megabit (Mb) = $2^{20}$ bits
1 Gigabyte (GB) = $2^{10}$ MB	1 Gigabit (Gb) = $2^{30}$ bits
1 Terabyte (TB) = $2^{10}$ GB	1 Terabit (Tb) = $2^{40}$ bits
1 Petabyte (PB) = $2^{10}$ TB	1 Petabit (Pb) = $2^{50}$ bits

## 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 ( $N_{16}$ )	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10

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

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

## Instrucciones vs. 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.

## Instrucciones vs. Datos (cont.)

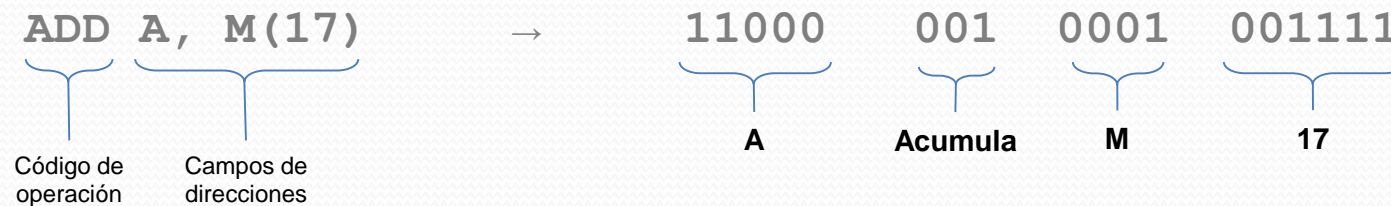
- Lenguaje natural:

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

- Lenguaje de programación de alto nivel:

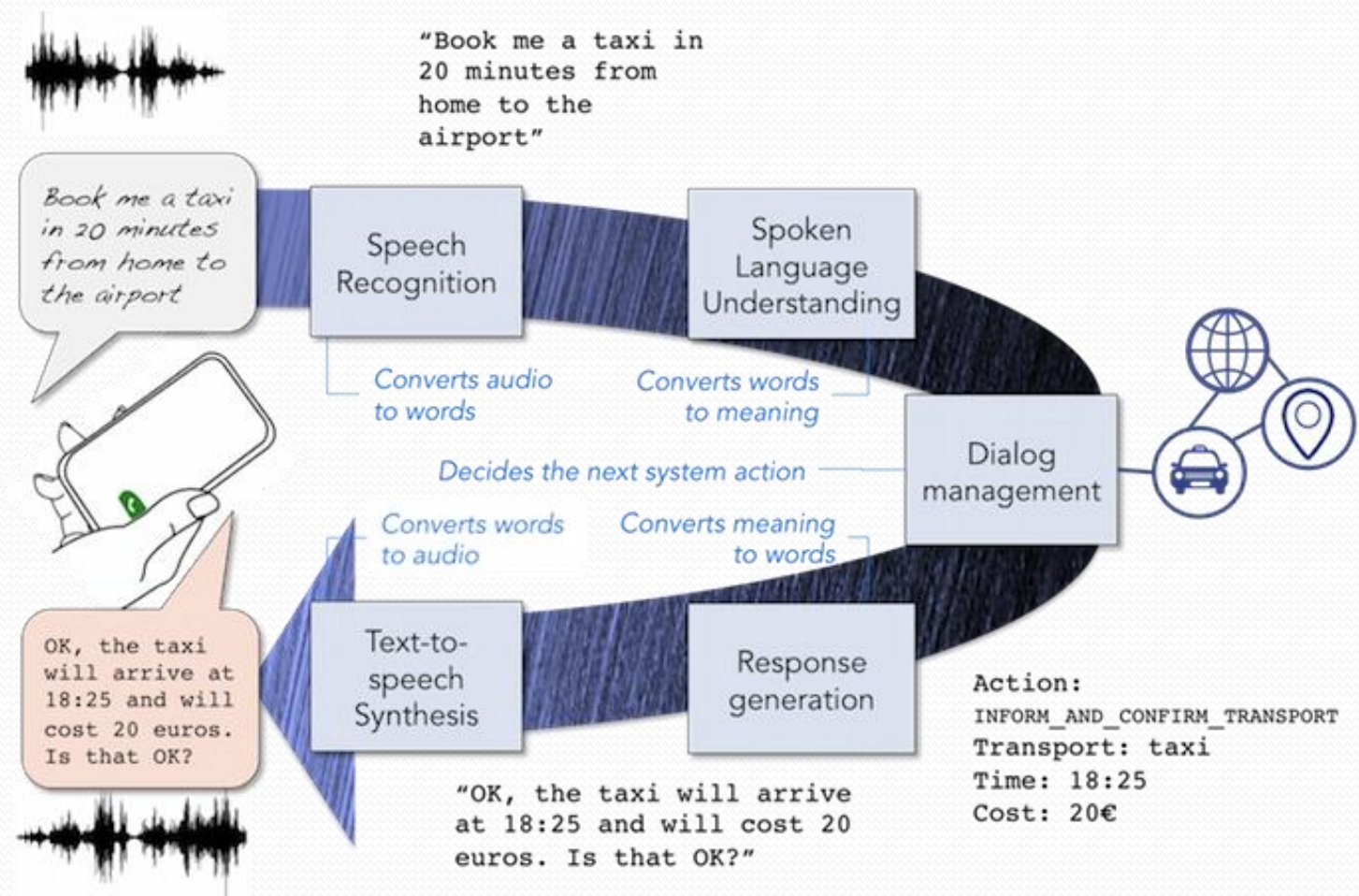
`A = A + M[17]`

- Ensamblador y lenguaje máquina:



## Instrucciones vs. Datos (cont.)

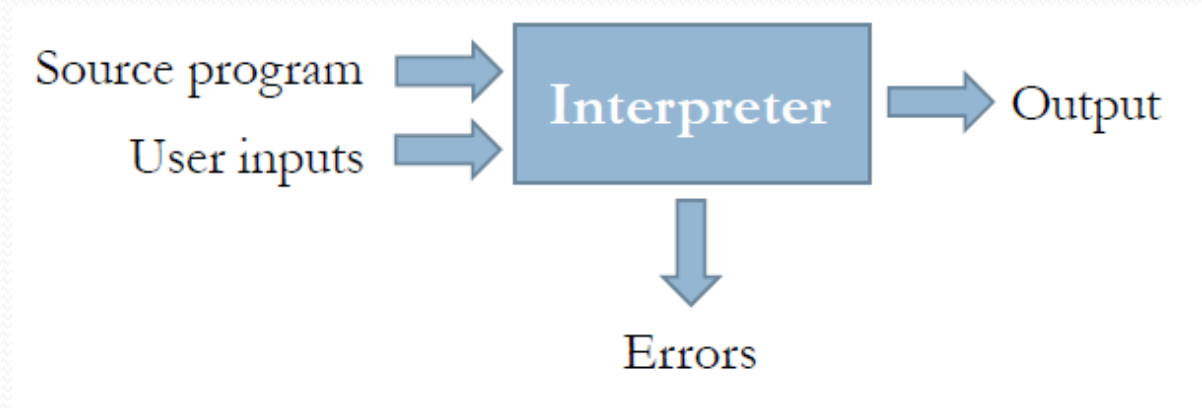
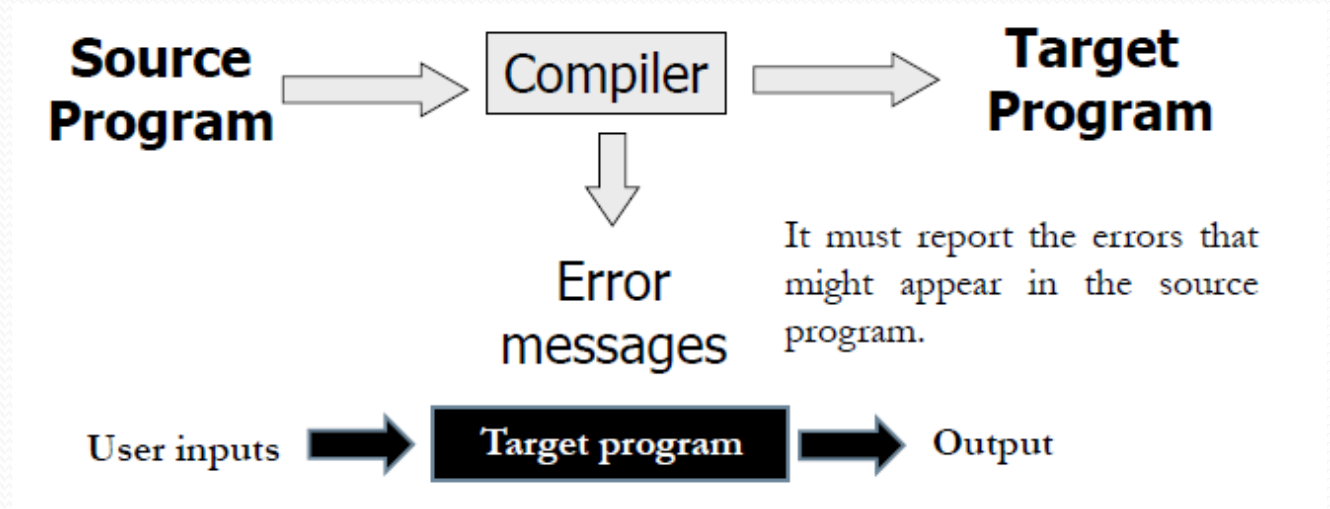
- Lenguaje natural:





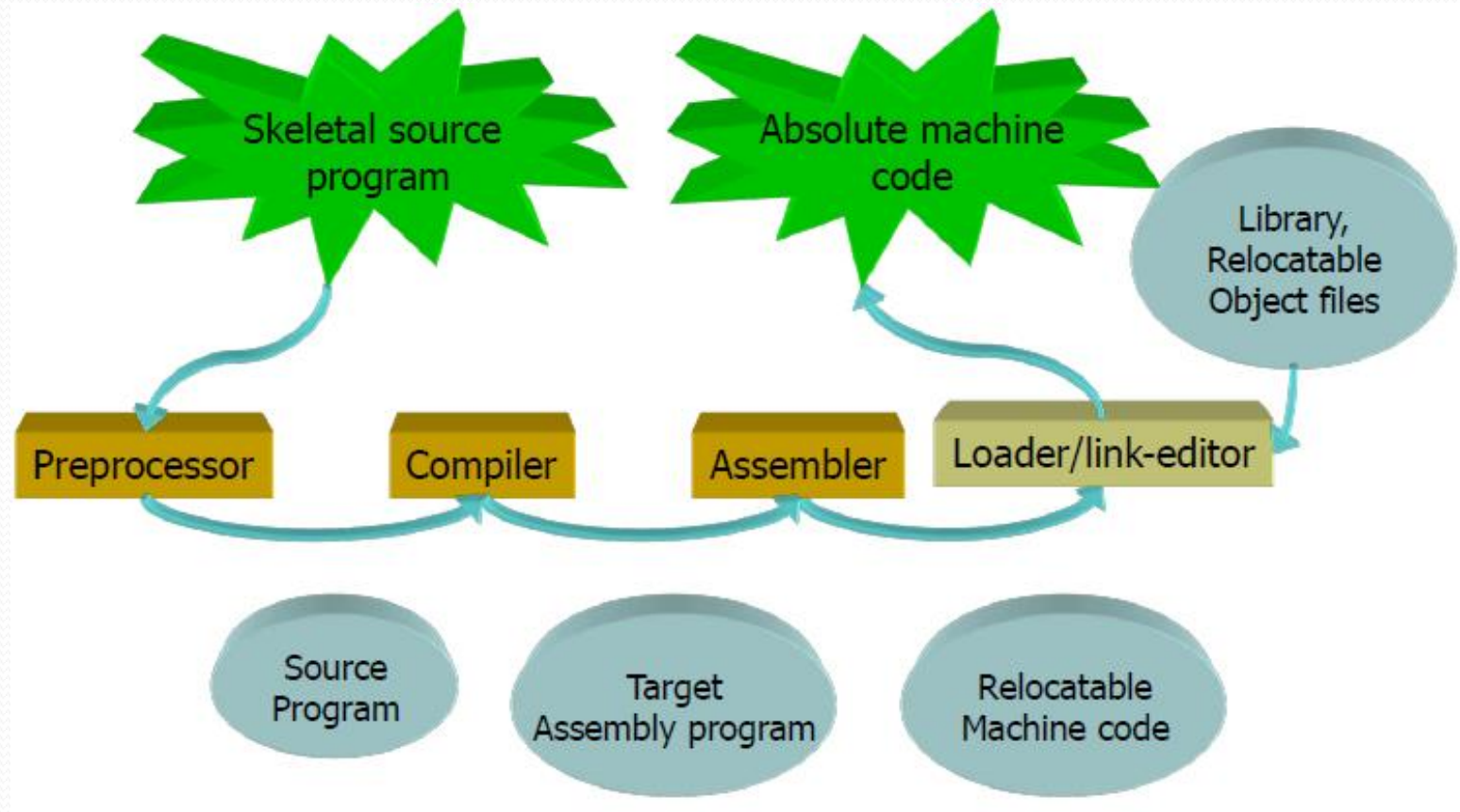
## Instrucciones vs. Datos (cont.)

- Lenguaje de programación de alto nivel:



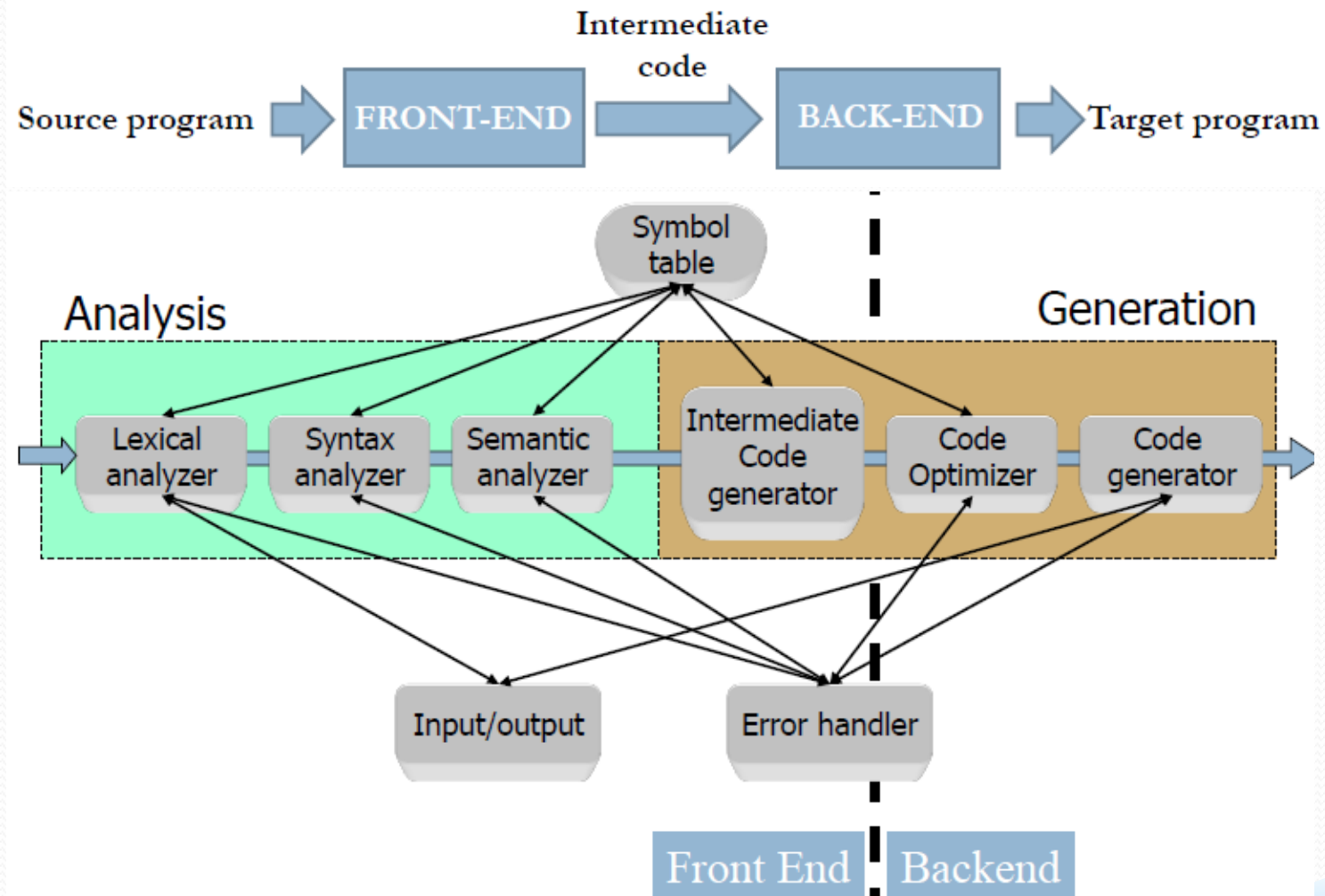
## Instrucciones vs. Datos (cont.)

- Lenguaje de programación de alto nivel:



## Instrucciones vs. Datos (cont.)

- Lenguaje de programación de alto nivel:



## Hardware (Soporte Físico)





## Firmware

Bloque de instrucciones de máquina para propósitos específicos grabado en una memoria, normalmente de lectura/escritura que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

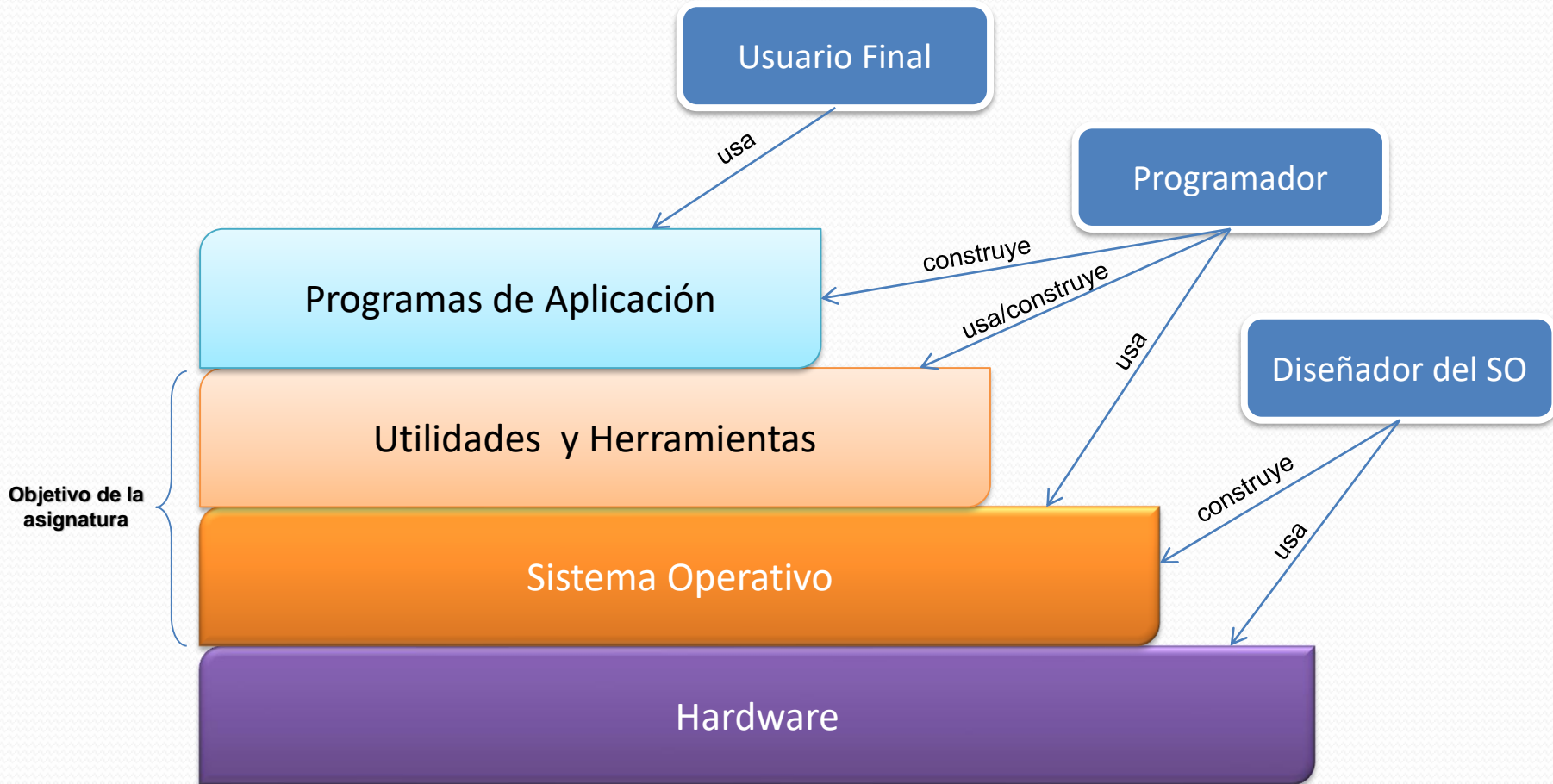


## Software (Soporte Lógico)

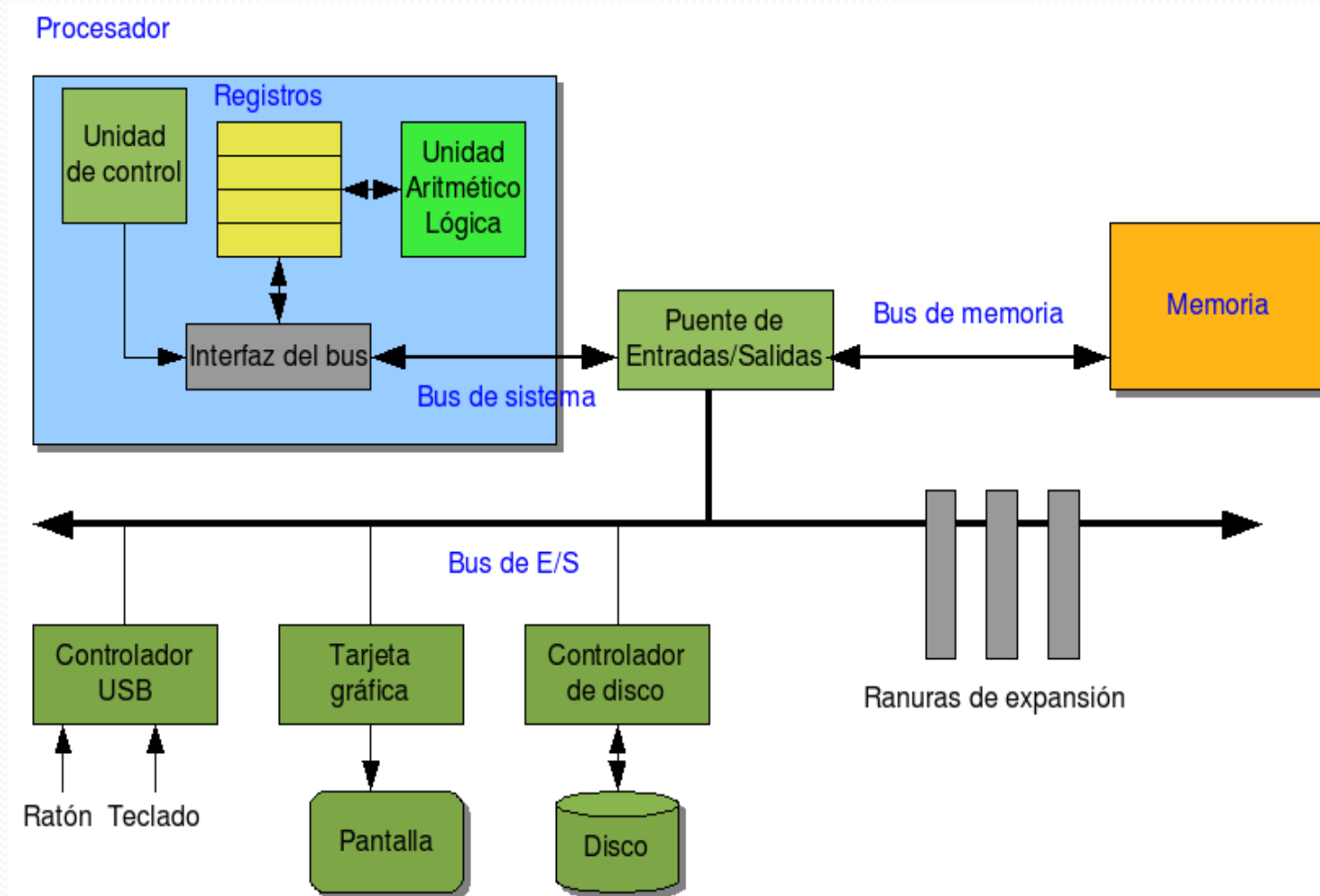
Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora (RAE).



## Definiciones Básicas [Stal05] (pp.55)



## Arquitectura de un Sistema

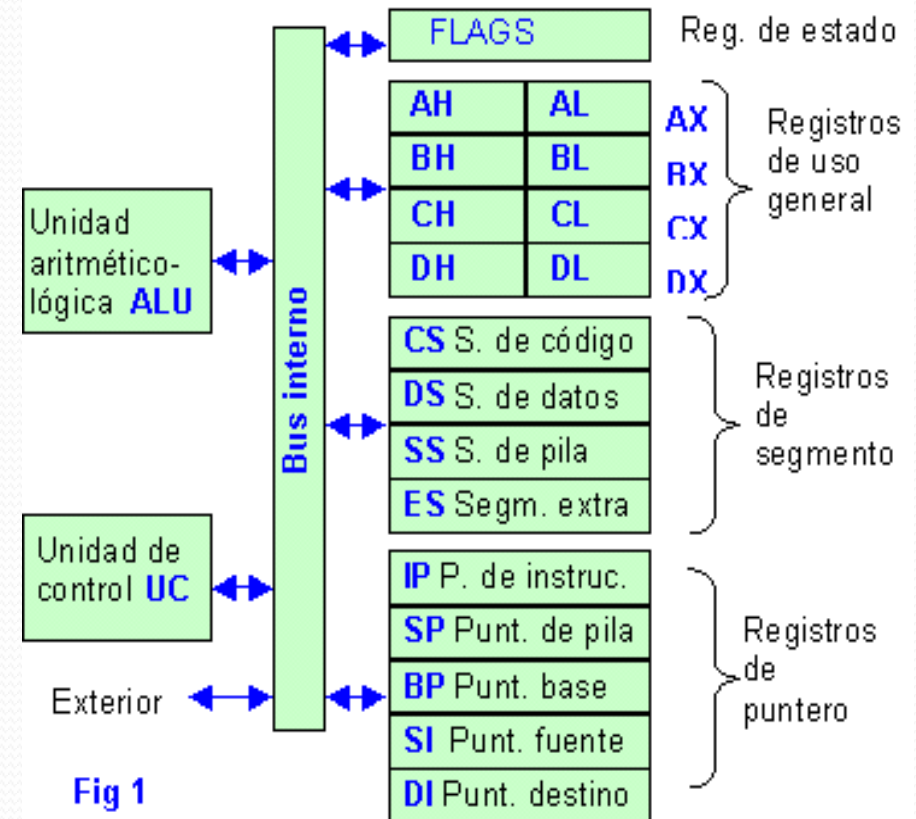




### Registros del Procesador [Stal05] (pp.11-13)

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

Esquema del microprocesador 8088



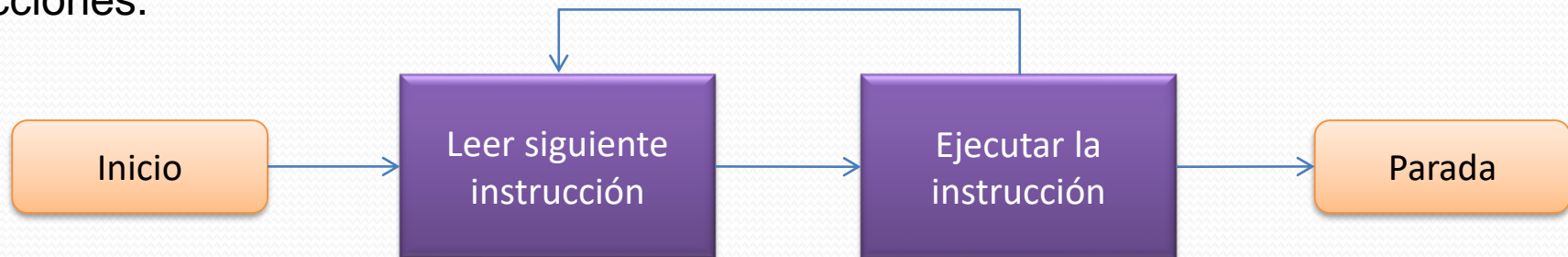
### Ejecución de Instrucciones [Stal05] (pp.14-17)

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.

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

La ejecución de un programa consiste en repetir el proceso de búsqueda y ejecución de instrucciones.



**Proceso** a seguir:

1. Se lee la instrucción cuya dirección está en el PC.
2. Se incrementa el PC en una unidad.
3. Se ejecuta la instrucción.

### Repertorio elemental de instrucciones [Stal05] (pp.14-17)

Podemos clasificarlas en:

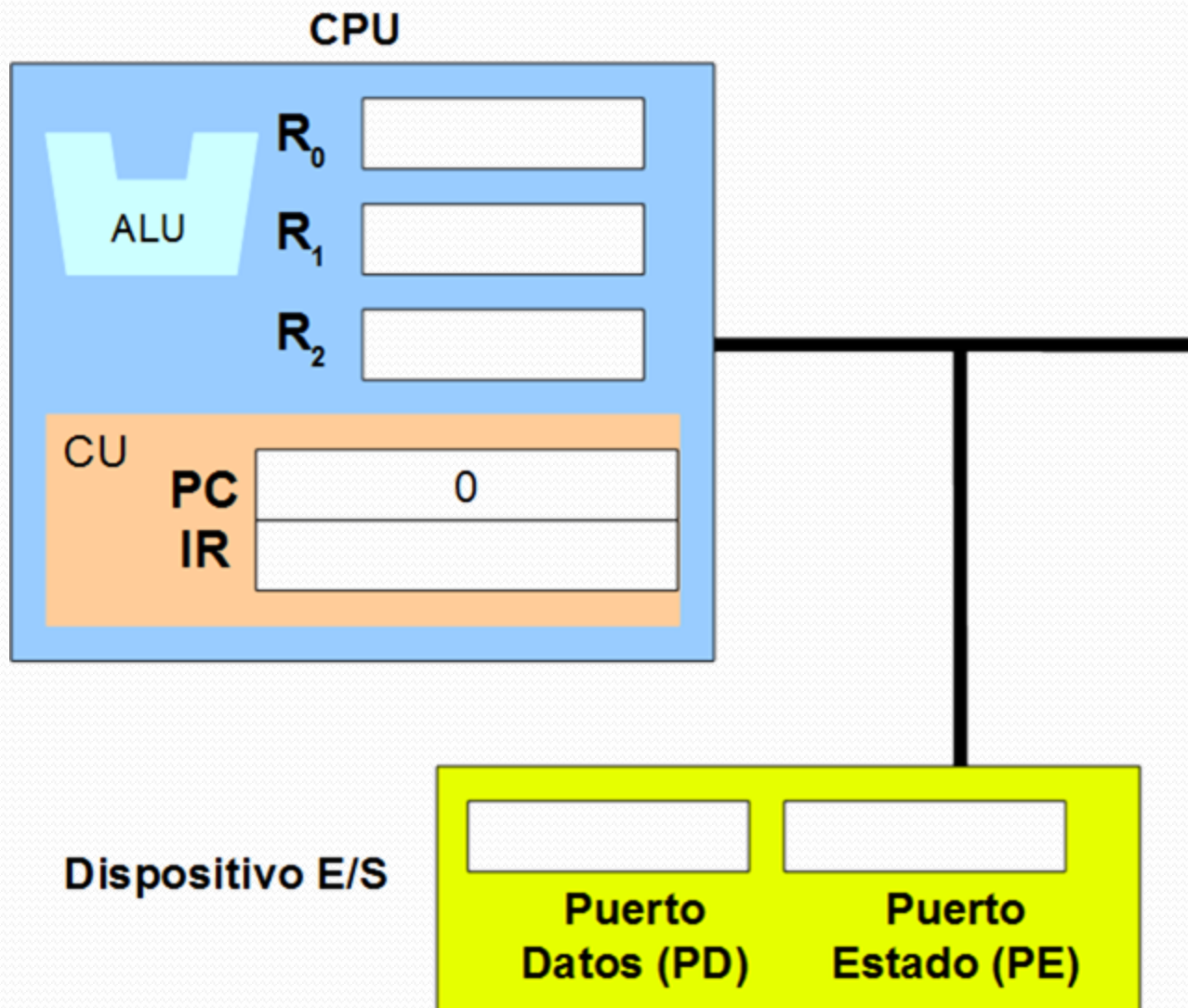
- **Transferencia de datos:**  
MOV origen, destino
- **Aritmético-lógicas:**  
ADD op1, op2, resultado  
COMP Ri, Rj  
MULT op1, op2, resultado  
...
- **Entradas/salidas:**  
IN puerto, destino  
OUT origen, puerto
- **Control:**  
JUMP/JNE/JE etiqueta  
CALL función  
RETURN e IRETURN  
HALT  
...

### Repertorio elemental de instrucciones [Stal05] (pp.14-17)

**Ejemplo:** Sumar dos números cargados en memoria en las direcciones direc1 y direc2 y poner el resultado en direc3.

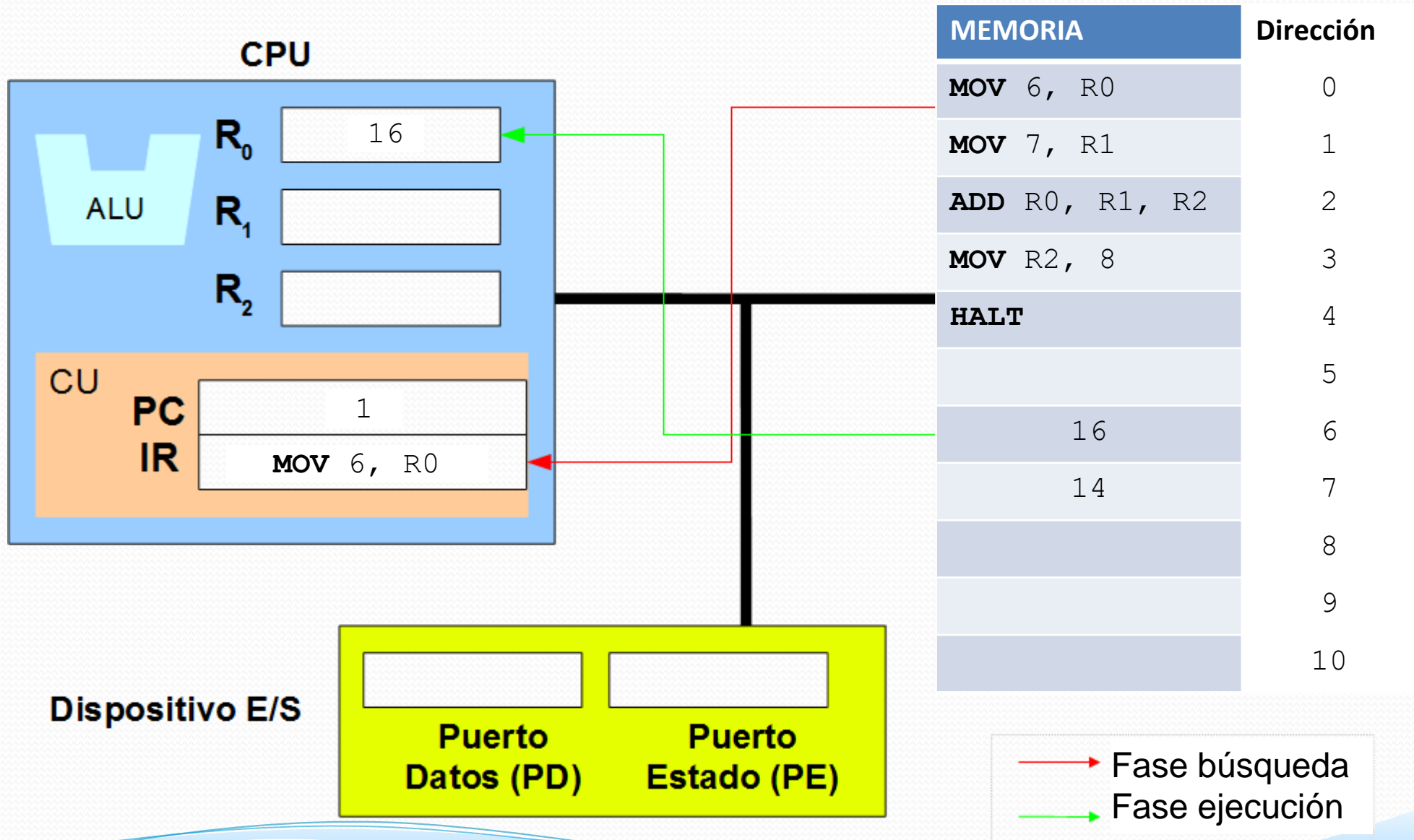
```
MOV direc1, R0
MOV direc2, R1
ADD R0, R1, R2
MOV R2, direc3
HALT
```

# Ejecución del ejemplo (paso 0)

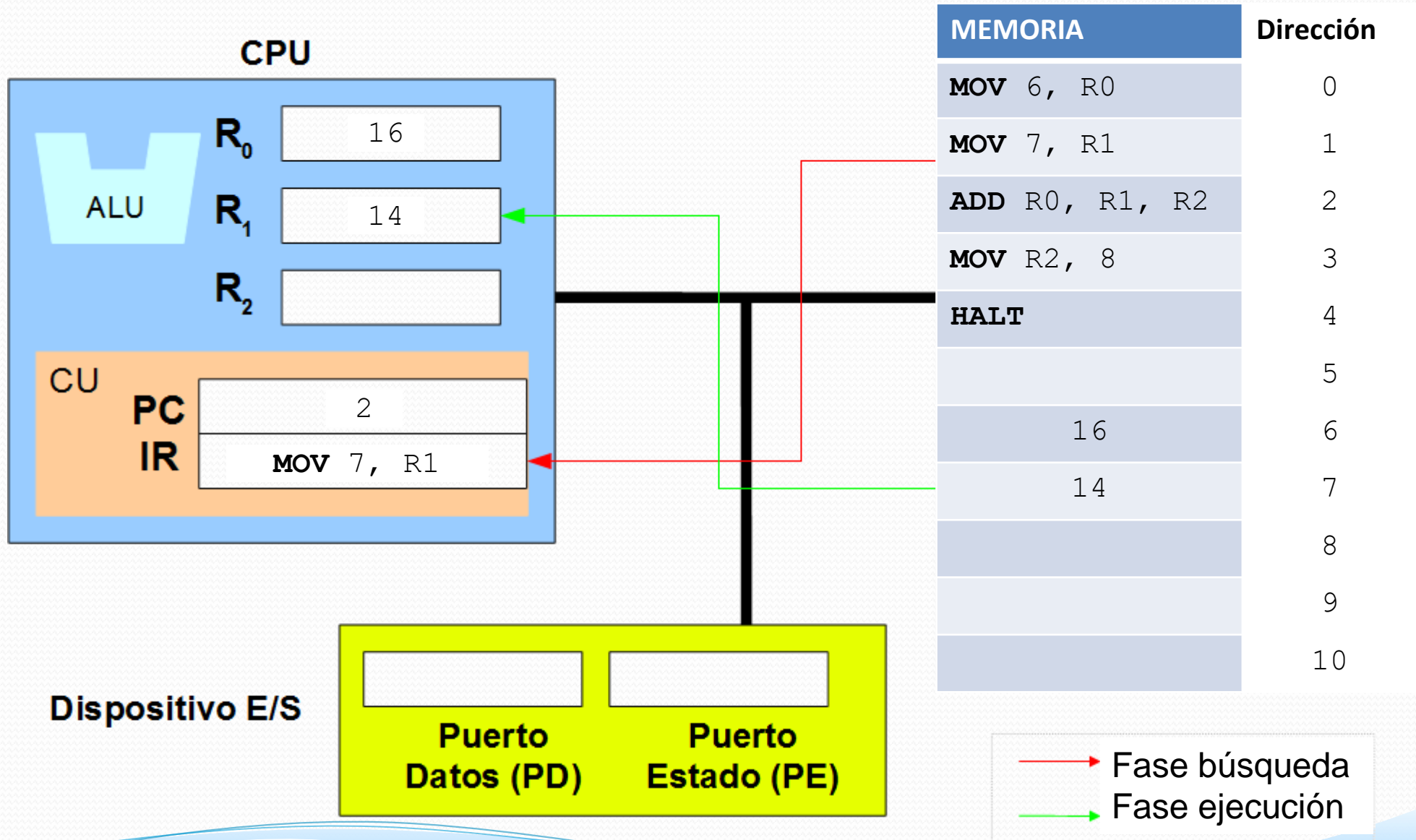


MEMORIA	Dirección
MOV 6, R0	0
MOV 7, R1	1
ADD R0, R1, R2	2
MOV R2, 8	3
HALT	4
	5
16	6
14	7
	8
	9
	10

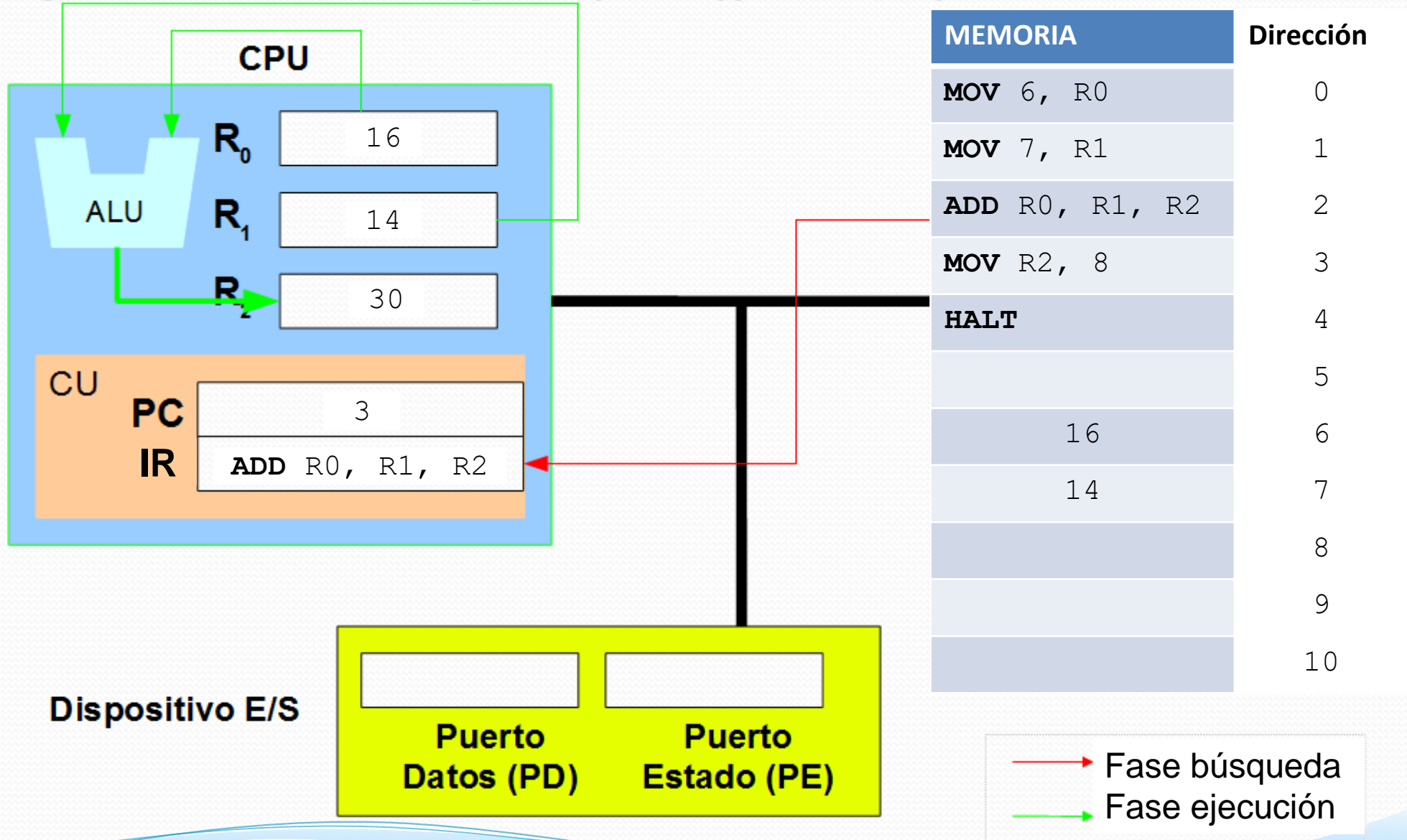
# Ejecución del ejemplo (paso 1)



# Ejecución del ejemplo (paso 2)

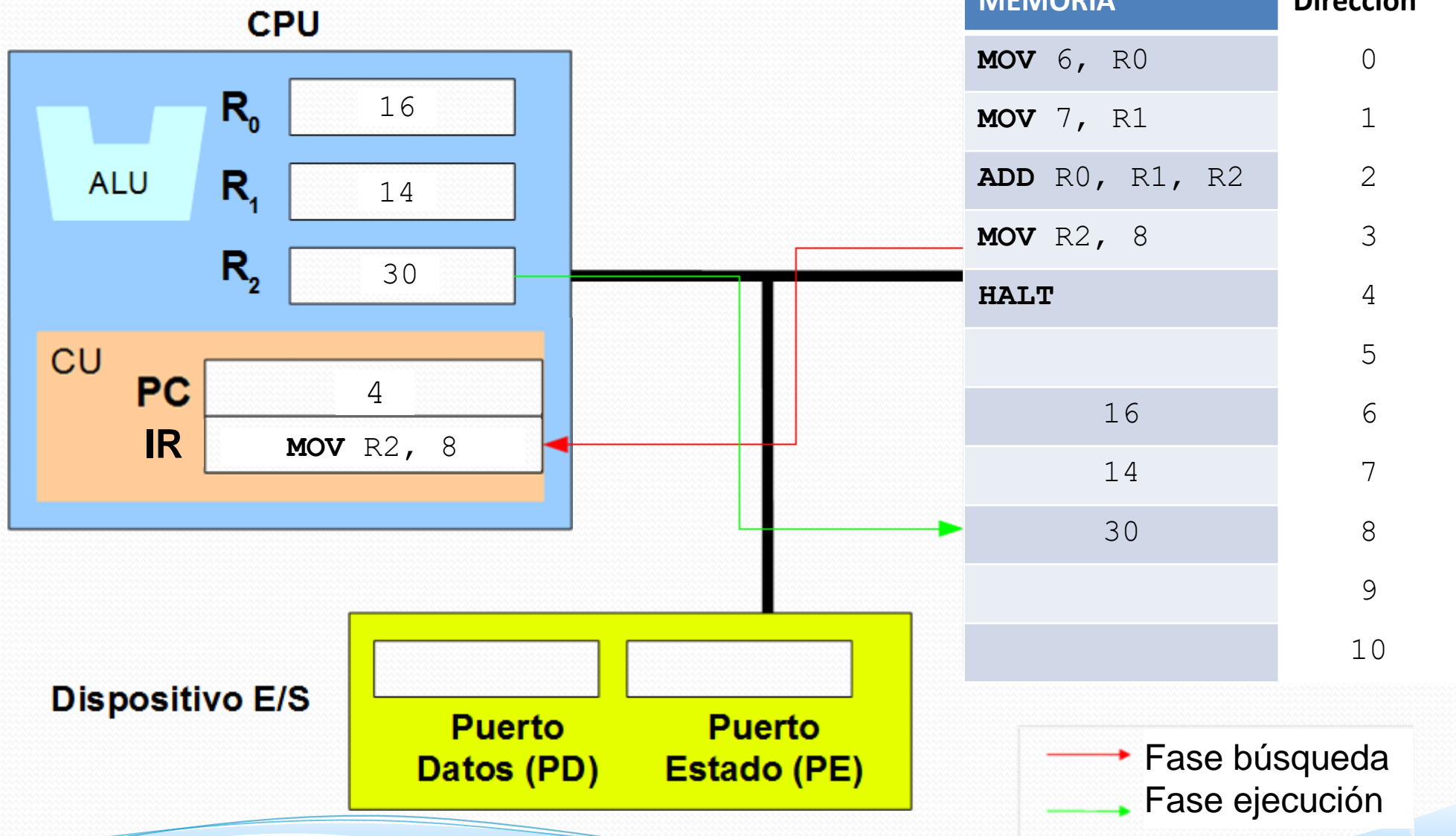


# Ejecución del ejemplo (paso 3)

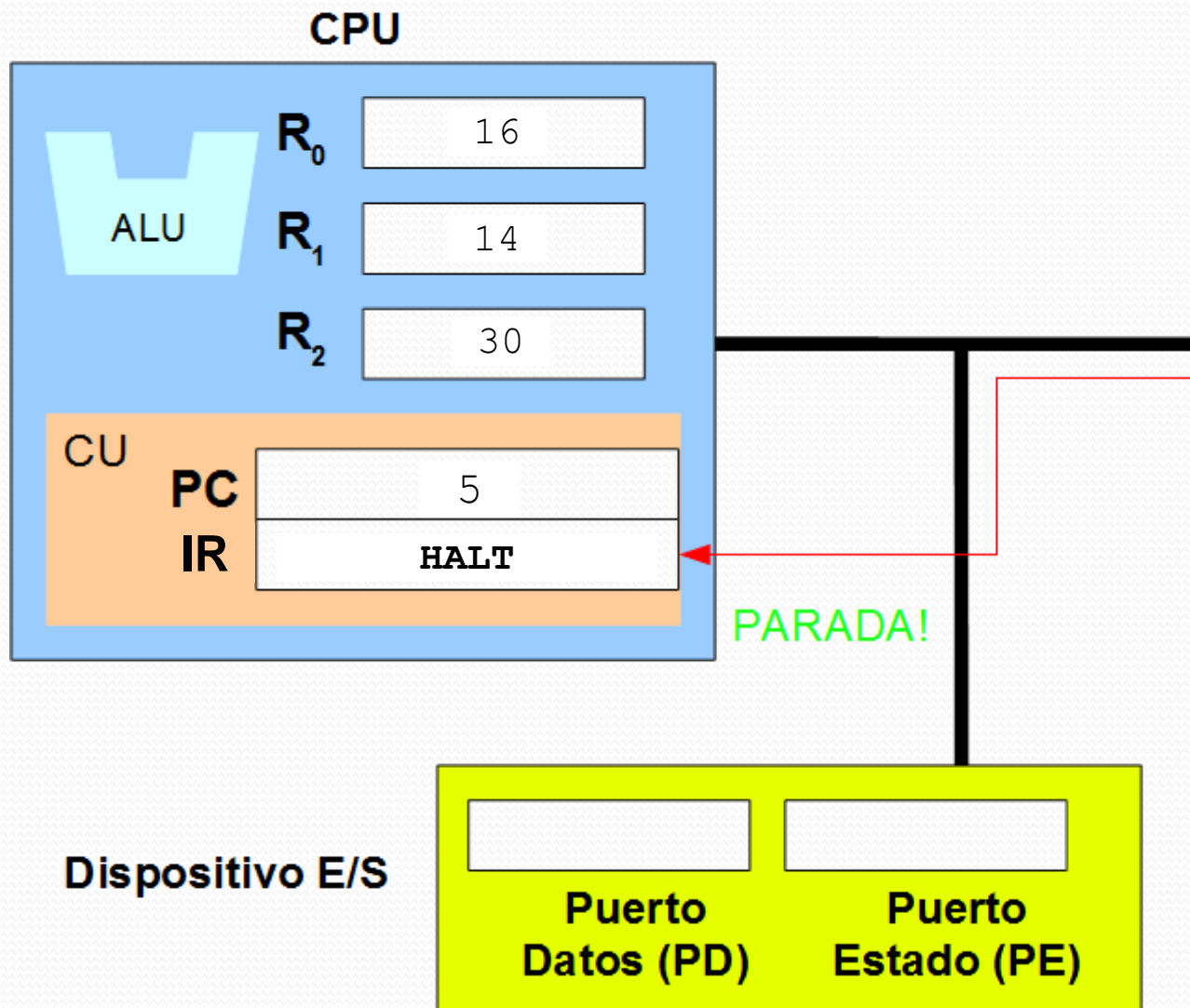




# Ejecución del ejemplo (paso 4)



# Ejecución del ejemplo (paso 5)



MEMORIA	Dirección
MOV 6, R0	0
MOV 7, R1	1
ADD R0, R1, R2	2
MOV R2, 8	3
HALT	4
	5
16	6
14	7
30	8
	9
	10

→ Fase búsqueda  
→ Fase ejecución

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

• **Ejemplo:** Sumar los números de la posición 940 y 941 y dejar el resultado en la posición 941.

1. El contador del programa (PC) tiene el valor 300 (la dirección de la primera instrucción). Se lleva a IR la instrucción y se incrementa PC en una unidad.
2. Se ejecuta la instrucción. (Los primeros 4 bits -dígitos en hexadecimal- en el IR indican que el AC será cargado desde memoria; los siguientes 12 bits -tres dígitos en hexadecimal- indican la dirección, 940).
3. La siguiente instrucción (5941) será captada desde la dirección 301. El PC se incrementa.
4. El anterior contenido del AC y el contenido de la dirección 941 se suman y el resultado se almacena en el AC.
5. La siguiente instrucción (2941) será captada desde la dirección 302. El PC se incrementa.
6. El contenido del AC se aloja en la dirección 941.

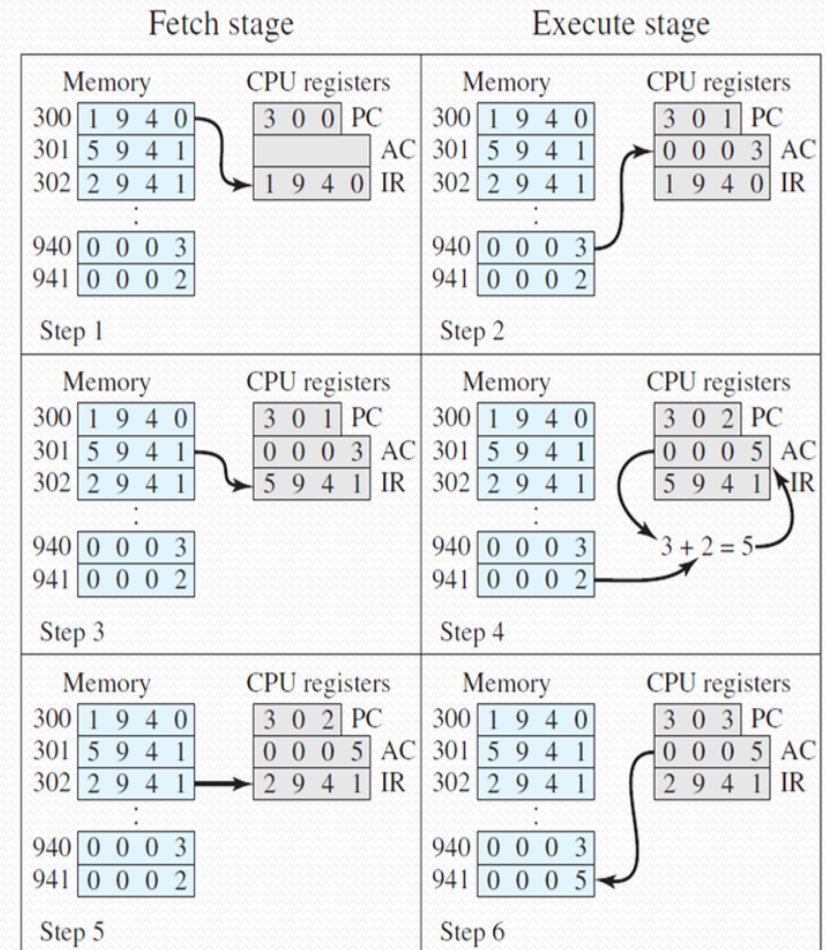


Imagen obtenida de [Stal05] pp.16

### Ejecución de Instrucciones [Stal05] (pp.34-37)

¿Qué sucede cuando un programa desea comunicarse con algún dispositivo de E/S?

#### → Comunicaciones de E/S

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

- E/S Programada.
- E/S Dirigida por Interrupciones.
- Acceso Directo a Memoria (Direct Memory Access, DMA).

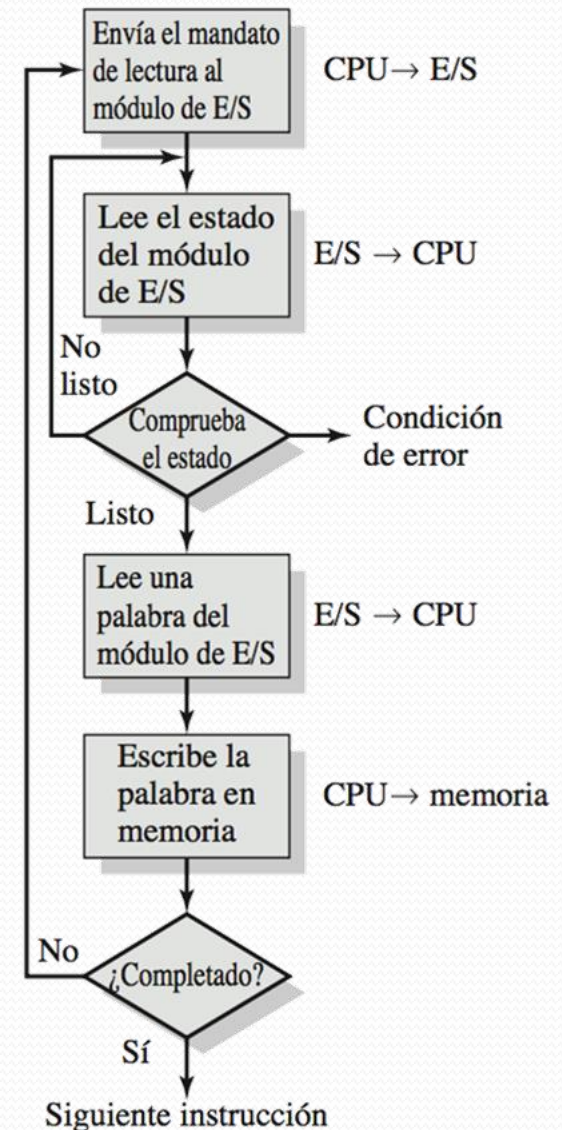
### Técnicas de Comunicación de E/S

- **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 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, se intenta que el procesador pueda continuar con trabajo útil.





## Técnicas de Comunicación de E/S

### • E/S Dirigida por Interrupciones.

Evento que interrumpe el flujo normal de ejecución y que está 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.

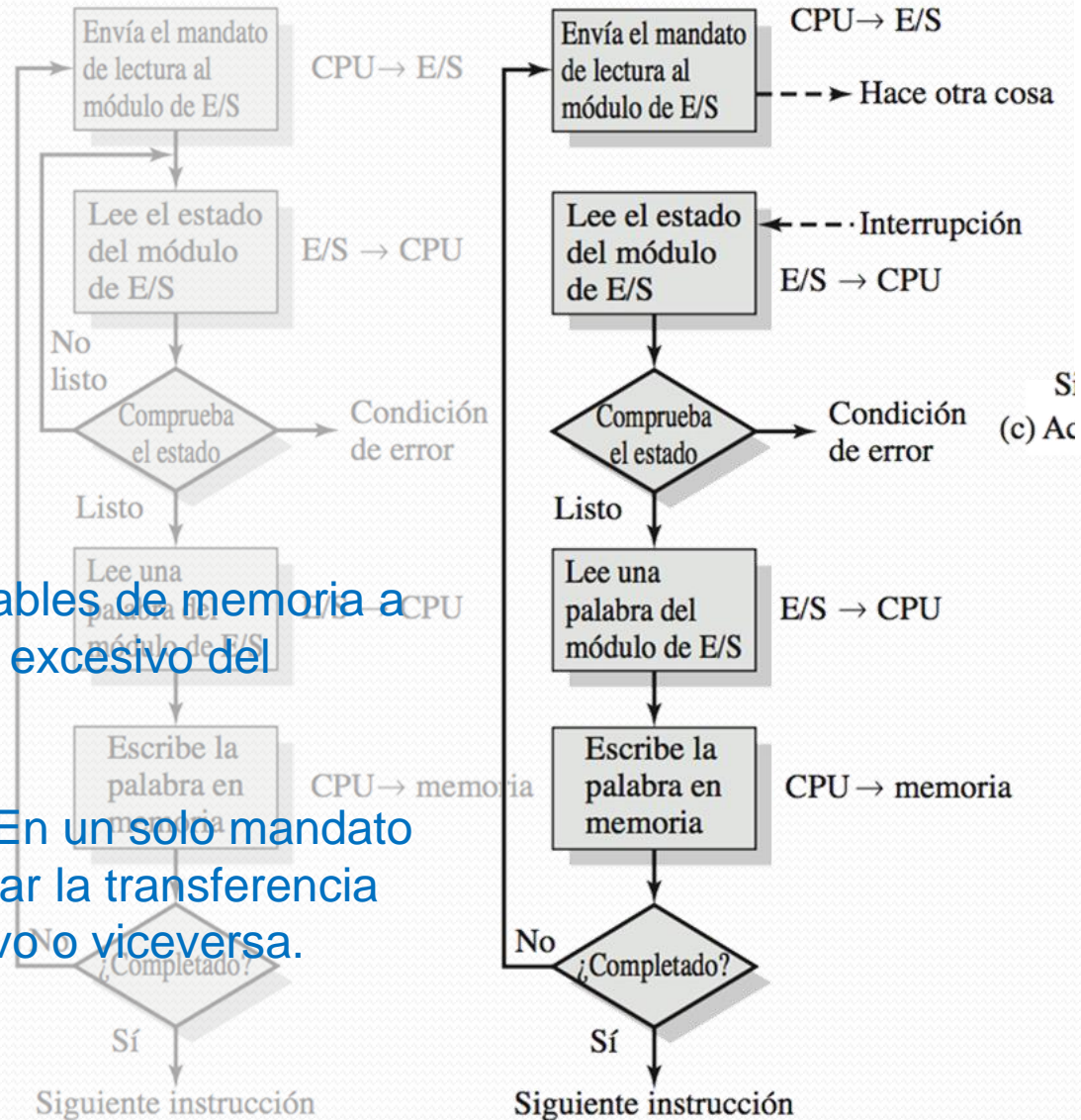
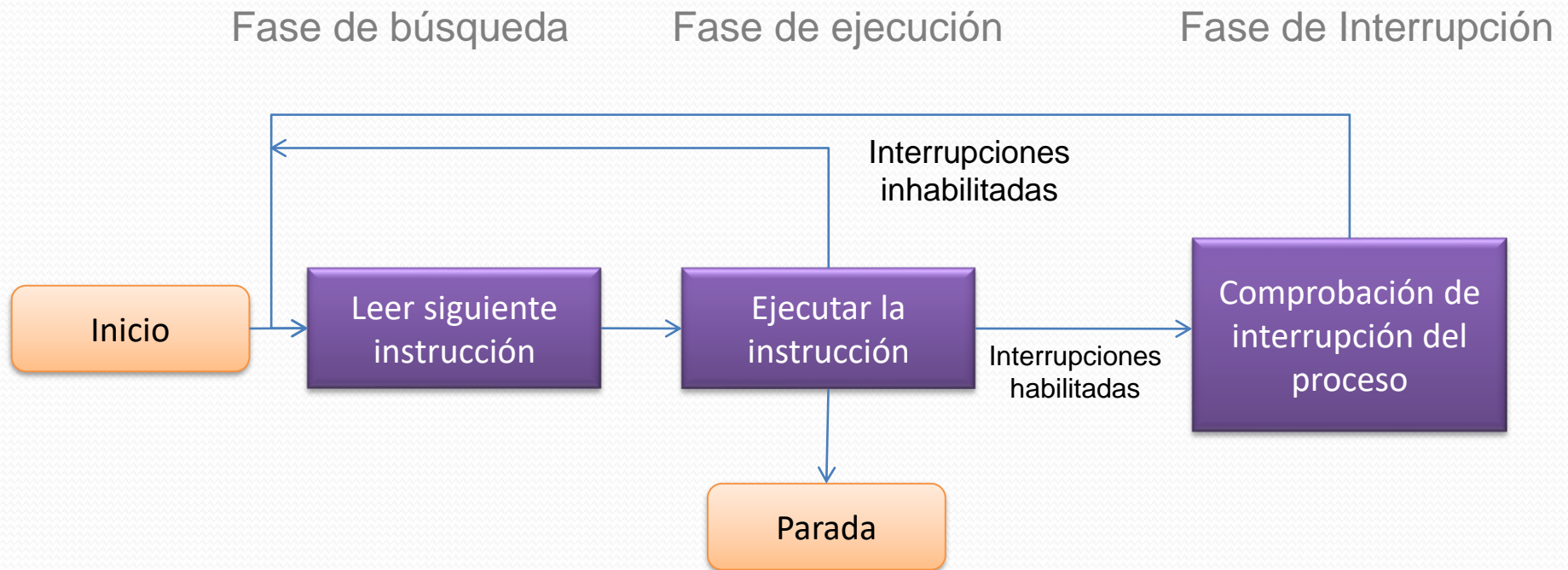


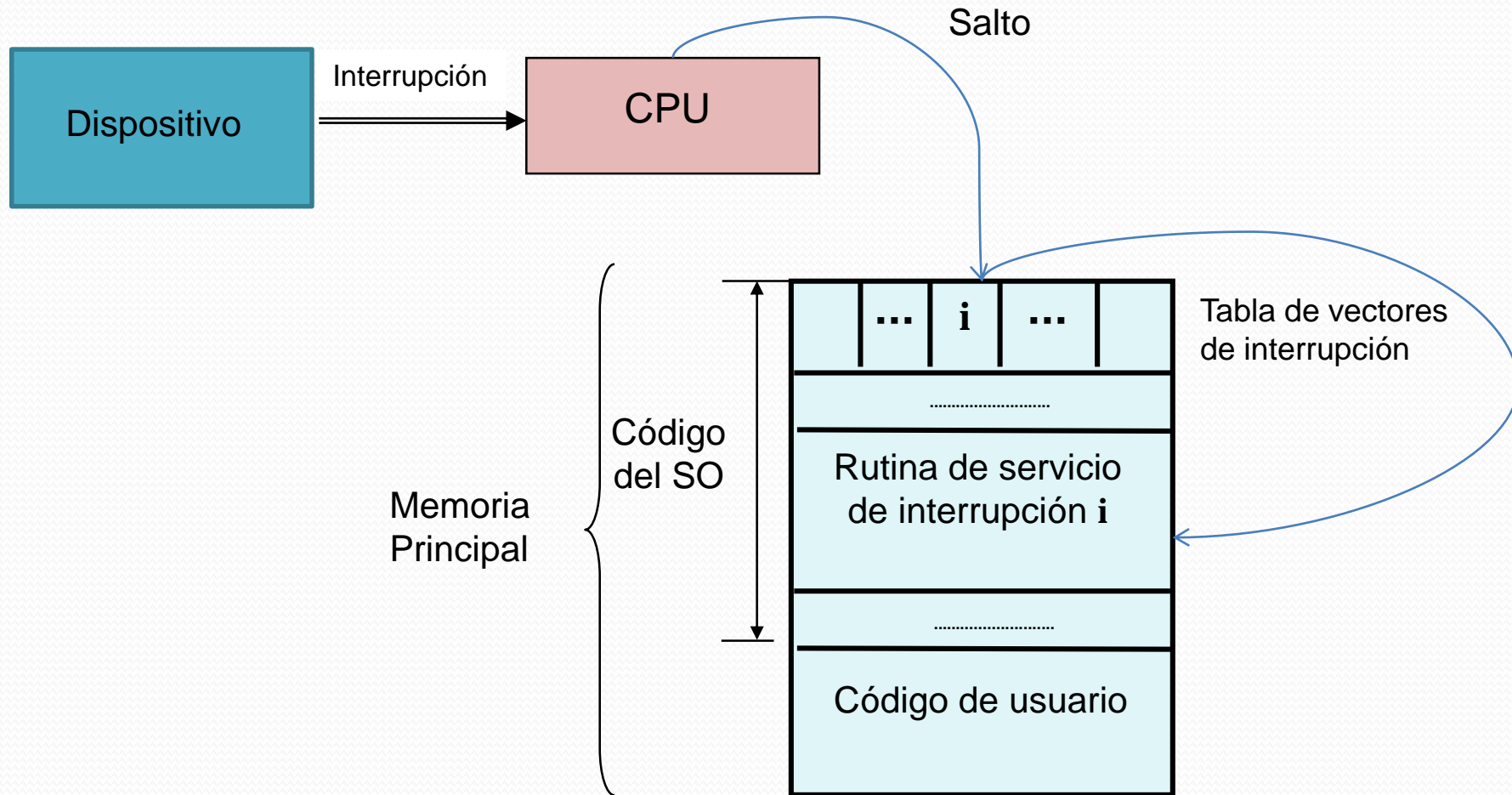
Imagen obtenida de [Stal05] pp.35

### Técnicas de Comunicación de E/S

- **Ciclo de instrucción** con interrupciones.

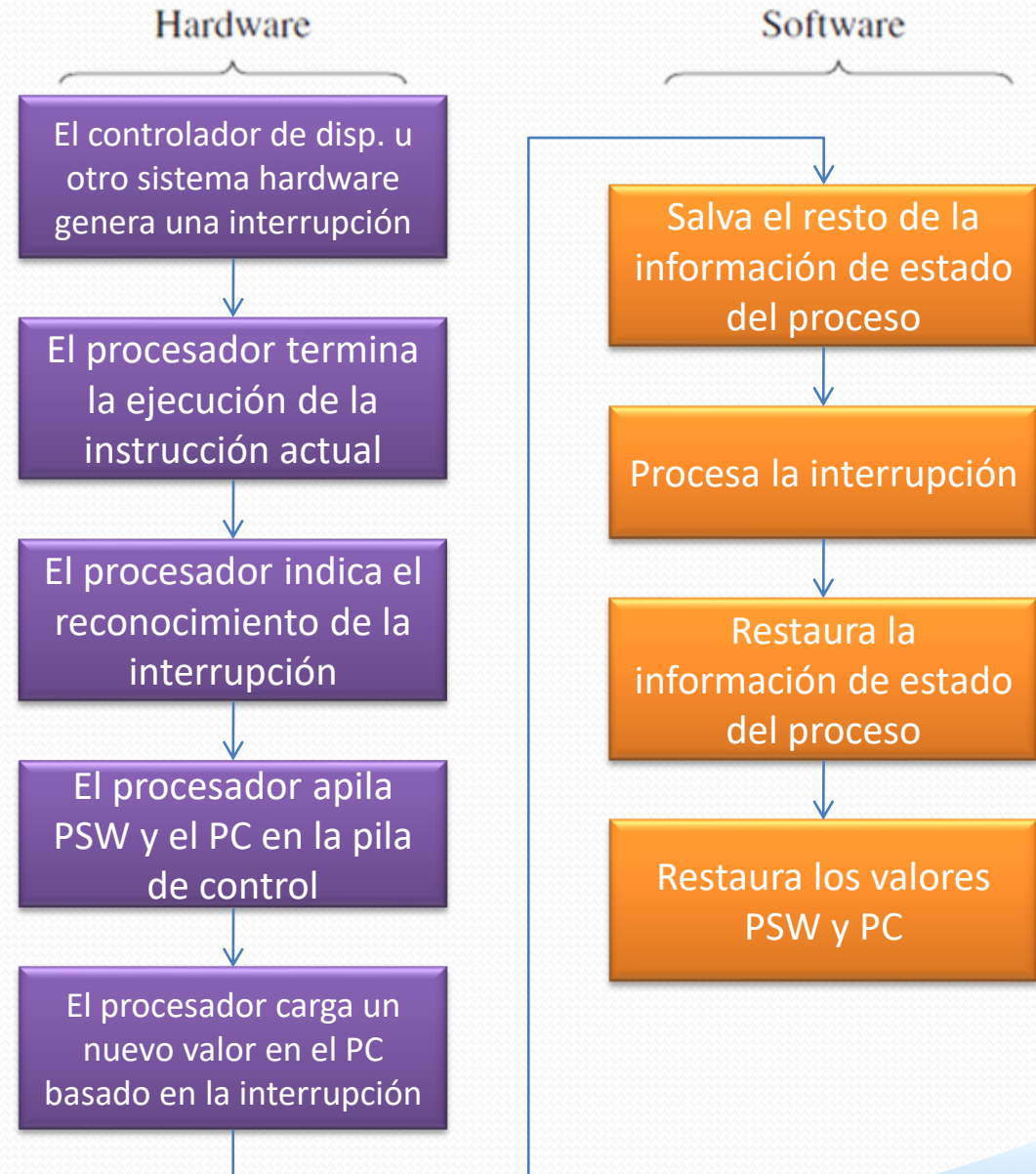


### Tratamiento de Interrupciones Vectorizadas





### Tratamiento de Interrupciones Vectorizadas [Stal05] (pp. 23. Fig. 1.10)



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

- **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.

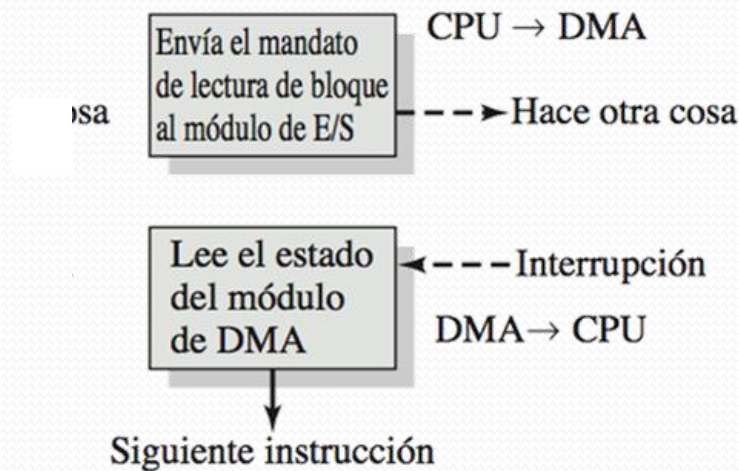


Imagen obtenida de [Stal05] pp.35

### Excepciones [Stal05] (pp.34-37)

- **Definición de excepción:** Evento inesperado generado por alguna condición que ocurre durante la ejecución de una instrucción (ejemplo, desbordamiento aritmético, dirección inválida, instrucción privilegiada, etc.). Es un evento síncrono.

### Protección del procesador [Carr07] (pp.26)

**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).

### Protección de los Dispositivos de E/S [Carr07] (pp.27-28)

- 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 [Carr07] (pp.26-27)

- 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).

### El Sistema Operativo [Stal05] (cap.2, pp.53-120)

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.



### El SO como interfaz Usuario/Computadora

Presenta al usuario una máquina abstracta más fácil de programar que el hardware subyacente:

- Oculta la complejidad del hardware.
- Da tratamiento homogéneo a diferentes objetos de bajo nivel (archivos, procesos, dispositivos, etc.).

Una aplicación se puede expresar en un lenguaje de programación y la desarrolla un programador de aplicaciones.

Es más fácil programar las aplicaciones en lenguajes de alto nivel que en el lenguaje máquina que entiende el hardware.



### El SO como interfaz Usuario/Computadora

Un SO proporciona normalmente utilidades en las siguientes áreas:

- **Desarrollo de programas** (editores de texto, compiladores, depuradores de programas).
- **Ejecución de programas** (cargador de programas y ejecución de éstos).
- **Acceso a dispositivos de E/S** (cada dispositivo requiere su propio conjunto de instrucciones).

### El SO como interfaz Usuario/Computadora

(cont.)

- **Acceso al sistema** (En sistemas compartidos o públicos, el SO controla el acceso y uso de los recursos del sistema: Shell, Interfaz gráfico).
- **Detección y respuesta a errores** (tratamiento de errores a nivel software y hardware).
- **Contabilidad** (estadísticas de uso de los recursos y medida del rendimiento del sistema).

### El SO como Administrador de Recursos

Un computador es un conjunto de recursos y el SO debe gestionarlos y para ello posee un mecanismo de control que cubre dos aspectos:

- Las funciones del SO actúan de la misma forma que el resto del software, es decir, son programas ejecutados por el procesador.
- El SO frecuentemente cede el control y depende del procesador para volver a retomarlo.

### El SO como Administrador de Recursos

Por lo tanto:

- El SO dirige al procesador en el uso de los recursos del sistema y en la temporización de la ejecución de otros programas.
- Una parte del código del SO se encuentra cargado en la memoria principal (kernel y, en ciertos momentos, otras partes del SO que se estén usando). El resto de la memoria está ocupada por programas y datos de usuario.
- La asignación de la memoria principal la realizan conjuntamente el SO y el hardware de gestión de memoria del procesador.
- El SO decide cuándo un programa en ejecución puede usar un dispositivo de E/S y también el acceso y uso de los ficheros. El procesador es también un recurso.

### Características deseables en un Sistema Operativo

- **Comodidad en el uso.**
- **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.

### 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.

### 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.
- ...