



# Tema 1 Introducción. Computadores y programas.

## *Contenido*

- 1 La computadora y sus componentes
- 2 Código máquina y código ensamblador
- 3 Lenguajes de alto nivel

# 1 La computadora y sus componentes

**Informática:** del francés “*information*”+”*automatique*”

Clasificación histórica:

**Microcomputadoras (PC)**

(actualmente en desfase)

**Minicomputadoras**

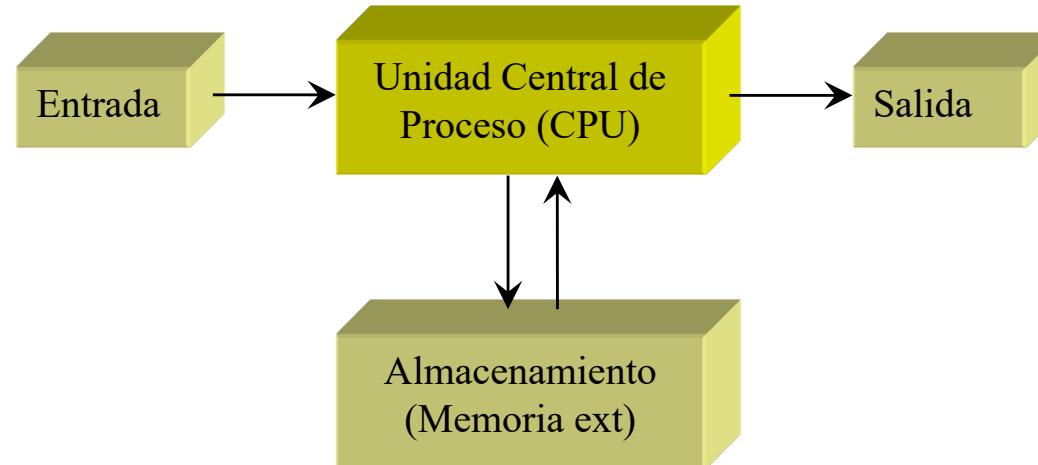
**Macrocomputadoras**

# 1.1 Computadoras

Componentes:

- Dispositivos de entrada
- Dispositivos de salida
- Dispositivos de almacenamiento
- Unidad de Procesamiento

} **Periféricos**



La Unidad de Proceso junto a la Memoria Central forman la **Unidad Central de Proceso (CPU)**

## 1.2 Programas

- Conjunto ordenado de instrucciones que indican a la computadora las acciones específicas que debe realizar para cumplir una tarea.
- Con un programa le decimos a la computadora **qué** tiene que hacer, el **orden** en el que lo tiene que hacer y qué **datos** usar para ello.
- Los programas han de estar escritos en un **lenguaje**:  
*Lenguaje Máquina* (lenguaje nativo de la máquina)  
*Lenguajes de alto nivel* (C, Pascal, etc...)

## 2 Código máquina y código ensamblador

### 2.1 Representación de la información

- La información se representa por ausencia/presencia de corriente (naturaleza eléctrica)

	Estado 0	Estado 1	
Impulsos eléctricos	_____		

- Unidades de información:

**Bit:** unidad mínima de información

**Byte:** agrupación de 8 bits (256 combinaciones)

**Palabra:** agrupación de 8, 16, 32 o 64 bytes

## 2.2 Programación de la computadora

- La computadora por sí misma (**hardware**) no realiza ninguna tarea. Necesita de los programas (**software**) para saber qué tiene que hacer.
- Definiciones:
  - Instrucción:** cadena de ceros y unos que especifican al procesador una operación elemental a realizar.
  - Lenguaje máquina:** conjunto de instrucciones que puede ejecutar un procesador y sus reglas de codificación.
- El lenguaje máquina difiere de unas computadoras a otras

## 2.3 Lenguaje máquina

**Lenguaje máquina:** Lenguaje en código binario que la computadora puede procesar directamente.

Etapas para realizar un programa en lenguaje máquina:

- Descomponer** el problema en operaciones elementales
- Asignar** un lugar en la memoria para cada instrucción y dato
- Codificar** cada instrucción como una serie de dígitos binarios según la tabla de instrucciones, máquina del procesador.

## 2.3 Lenguaje máquina

Ejemplo: Computadora imaginaria con instrucciones de 8 bits, donde los 3 primeros son el código de operación y los 5 siguientes la dirección del operando en memoria.

Código	Operación	Descripción
001	Cargar	acumulador = operando
010	Sumar	acumulador = acumulador+operando
011	Restar	acumulador = acumulador-operando
100	Bifurcar	Si acumulador = 0 salta a la dirección del operando.
101	Comparar	Si acumulador = operando entonces acumulador=0
110	Almacenar	operando = acumulador
111	Parar	Detiene la ejecución del programa.

## 2.3 Lenguaje máquina

El siguiente programa realiza la resta entre dos números:

Código	Operación
001	Cargar
010	Sumar
011	Restar
100	Bifurcar
101	Comparar
110	Almacenar
111	Parar

Memoria Central		
Dirección	Instrucción	Contenido
00000	<u>00100101</u>	Programa
00001	<u>01100110</u>	
00010	<u>11000100</u>	
00011	<u>11100000</u>	
00100	00000000	Datos
00101	00000111	
00110	00000010	

## 2.4 Lenguaje ensamblador

El lenguaje ensamblador surge para subsanar los problemas del código máquina.

Utiliza **nemotécnicos** para los nombres de instrucción y etiquetas para los datos en memoria.

Cada instrucción en ensamblador tiene su correspondiente instrucción máquina.

Código	Abreviatura	Operación
001	CAR	Cargar
010	ALM	Almacenar
011	SUM	Sumar
100	RES	Restar
101	BC	Bifurcar si es cero
110	BNC	Bifurcar si no es cero
111	PAR	Parar

## 2.4 Lenguaje ensamblador

El programa anterior en ensamblador sería:

CAR	A
RES	B
ALM	C
STOP	
C:	0
A:	7
B:	2

## 3 Lenguajes de Alto Nivel

- El lenguaje ensamblador no es portable. Además sigue siendo demasiado cercano a la máquina
- Los **Lenguajes de Alto Nivel** utilizan gramáticas más naturales, lo cual facilita el proceso de programación.
- Mediante el proceso de **compilación** (traducción a lenguaje máquina) pueden ser ejecutados en cualquier máquina.
- Ejemplo de instrucción en COBOL:  
SUBSTRACT I MPUESTOS FROM SUELDO GIVING LIQUIDO  
o también:  
LIQUIDO = SUELDO-IMPUESTOS
- Una instrucción en un lenguaje de alto nivel equivale a varias instrucciones en lenguaje máquina

## 3 Lenguajes de Alto Nivel

---

Ventajas de los Lenguajes de Alto nivel:

- Mayor productividad
- Más fáciles de aprender
- Autodocumentados. Las variables tienen un nombre que imponemos dependiendo del contexto.
- No hay que conocer la máquina
- Son independientes de máquina (portables).

## 3.2 Clasificación de los LP's

Según el paradigma hablaremos de:

- Imperativo** o procedimental (ej: C, Pascal, etc.)
- Declarativo**, que a su vez se divide en funcional o aplicativo (ej:Lisp) y lógico (ej: Prolog)
- Orientado a objetos** (ej: Java, C++, etc.)

Y según el proceso de traducción:

- Lenguajes compilados**
- Lenguajes interpretados**

En este curso nos centraremos en el lenguaje **C**, que sigue el paradigma imperativo y es, básicamente, compilado.