

# 1. Introducción al computador

## Informática

*Ingeniería en Electrónica y Automática Industrial*

RAÚL DURÁN DÍAZ    JUAN IGNACIO PÉREZ SANZ

Departamento de Automática  
*Escuela Politécnica Superior*

Curso académico 2023–2024

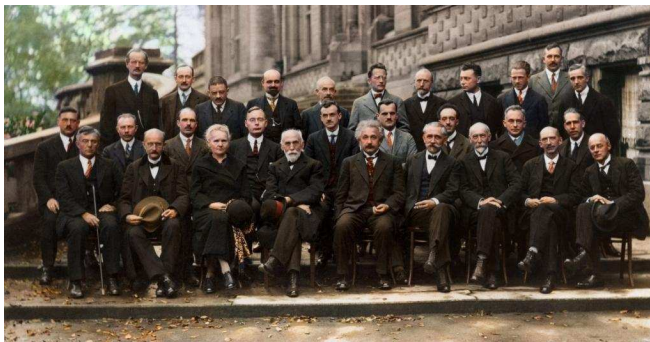
# Contenidos

- 1 Definiciones básicas
- 2 Estructura funcional
- 3 Evolución histórica

# ¡Y la Física tiene la culpa!

- Los sistemas de computación están en todas partes.
- Una revolución total e imprevista en 40 años.
- Todo basado en **la Física del Estado Sólido**.

# Aquí están los padres de nuestra tecnología:



SOLVAY CONFERENCE 1927

coloured by posthincolour.com

A. PICARD   E. HENRIOT   P. EHRENFEST   Ed. HERSEN   Th. DE DONDER   E. SCHRÖDINGER   E. VERSCHAFFELT   W. PAULI   W. HEISENBERG   R.H. FOWLER   L. BRÉLOUIN  
P. DEBYE   M. KNUDSEN   W.L. BRAGG   H.A. KRAMERS   P.A.M. DIRAC   A.H. COMPTON   L. de BROGIE   M. BORN   N. BOHR  
I. LANGMUIR   M. PLANCK   Mmo CURIE   H.A. LORENTZ   A. EINSTEIN   P. LANGEVIN   Ch.E. GUYE   C.T.R. WILSON   O.W. RICHARDSON  
Absents : Sir W.H. BRAGG, H. DISSLANDRES et E. VAN AUBEL

Figura: Conferencia Solvay, Bruselas, 1927

## Están en todas partes. . .

- La velocidad de computación ha crecido exponencialmente.
- Se han desarrollado también vertiginosamente las redes de comunicaciones, que proporcionan acceso casi ubicuo y alta velocidad de conexión.
- El diseño, desarrollo y fabricación de software representan un gran porcentaje del PIB en muchos países.
- Además, el precio de los sistemas se ha dividido muchas veces por 10.

⇒ La computación y la comunicación han permitido la tercera revolución de nuestra civilización: la Sociedad de la Información.

# Sociedad de la Información

- La activos importantes son intangibles.
  - Costosos de diseñar y depurar.
  - Fáciles de mover, transportar, replicar, coste casi cero.
- Las malas noticias:
  - La copia indiscriminada perjudica gravemente la industria.
    - Se hace necesario algún tipo de protección de la propiedad intelectual.
  - Nuevas amenazas en forma de ciberataques.
    - Se hace necesaria la protección de la información individual y de las infraestructuras críticas.

# Sociedad de la Información: la educación

## Sistemas de información en las aulas.

- Ventajas:
  - Herramientas cómodas para mejorar el aprendizaje.
  - Flexibles.
  - Contenido multimedia.
- Inconvenientes:
  - Evanescencia.
  - Imprecisión.
  - Tienen a eliminar la necesaria reflexión personal.

# ¿Qué instrumento soporta todo esto?

## Sistemas de información

Un **sistema de información** toma información, la procesa y devuelve la información transformada de acuerdo a un plan previsto.

Es como una fábrica de manufacturas, donde la materia prima es *información*.

- Almacenes: → memoria principal.
- Oficina técnica: → unidad de control.
- Líneas de producción: → ruta de datos y unidades funcionales.



# Algunas definiciones

## Definición

Se denomina **codificación** a una correspondencia biunívoca entre los elementos de un conjunto y los de otro.

## Observación

De esta manera, podemos referirnos a elementos del primer conjunto “representándolos” mediante los del segundo.

# Más definiciones

## Definición

**Computador:** máquina capaz de recibir unos *datos* de entrada, efectuar con ellos *operaciones aritméticas y lógicas*, y suministrar los resultados en forma de *datos de salida*, todo ello bajo el control de un programa de instrucciones previamente cargado en el propio computador.

# Más definiciones

## Definición

Un **dato** es un conjunto de uno o más símbolos que son aptos para representar alguna realidad, no necesariamente cuantitativa.

## Observación

Pueden ser datos: una temperatura, el nombre de una persona, el título de una obra de teatro. . . que deberemos *codificar* para representarlo en el computador.

# Más definiciones

## Definición

Una **instrucción** es un símbolo que representa una orden para el computador. Cada posible orden está codificada en una instrucción.

## Definición

Un **programa** es una lista secuencial de instrucciones. El computador ejecuta las instrucciones exactamente en el orden en que vienen marcadas en la lista.

## Observación

Algunas instrucciones sirven para modificar el orden secuencial de ejecución, estableciendo como siguiente instrucción otra distinta de la sucesiva según el programa. La decisión puede ser dinámica, es decir, depende del estado de la máquina a lo largo de la ejecución.

# Más definiciones

## Definición

Cualquiera de las codificaciones de las que hemos hablado usan en última instancia un código binario, en donde cada unidad elemental se llama **bit**. Al conjunto de 8 bits se le denomina **octeto** o (más frecuentemente) **byte**.

# Unidades y múltiplos

Se utiliza el símbolo  $b$  para representar 'bit' y el símbolo  $B$  para representar 'byte'.

Prefijo	Símbolo	Factor
Kilo	k-	$10^3$
Mega	M-	$10^6$
Giga	G-	$10^9$
Tera	T-	$10^{12}$
Exa	E-	$10^{15}$
Peta	P-	$10^{18}$

- 1 kB significa mil bytes.
- 5 Mb significa cinco millones de bits.

# Unidades y múltiplos

Se usa también el siguiente método para expresar múltiplos.

Prefijo	Símbolo	Factor
Kilo	Ki-	$2^{10}$
Mega	Mi-	$2^{20}$
Giga	Gi-	$2^{30}$
Tera	Ti-	$2^{40}$
Exa	Ei-	$2^{50}$
Peta	Pi-	$2^{60}$

- Por ejemplo, 1 kB significa mil bytes, mientras que 1 KiB significa 1024 bytes.
- A veces se usa KB como sinónimo de KiB: ¡atención a la K mayúscula!

# Antes de abrir la tapa. . .

- Dispositivos de entrada/salida (E/S):
  - teclado,
  - ratón,
  - pantalla.



# Lo que se ve. . .



Figura: Teclado

# Lo que se ve. . .



Figura: Ratón

# Lo que se ve. . .



Figura: Pantalla

## Abriendo la tapa...

- Placa madre (motherboard), con
  - procesador,
  - memoria,
  - buses de interconexión,
    - bus del sistema, EISA (Extended Industry Standard Architecture).
    - bus IDE (Integrated Drive Electronics) para discos,
    - bus de expansión de periféricos PCI (Peripheral Component Interconnect).
    - otros buses de E/S (SCSI, ...).

# Abriendo la tapa...

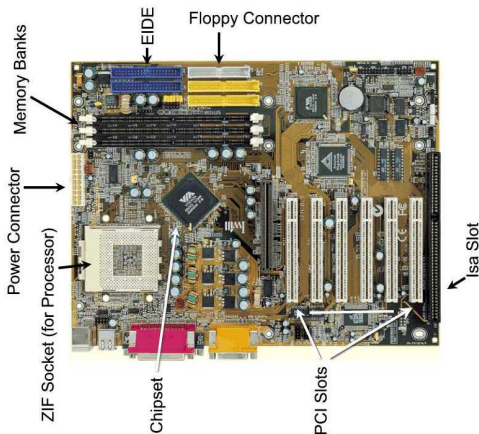


Figura: Placa madre

# Abriendo la tapa...

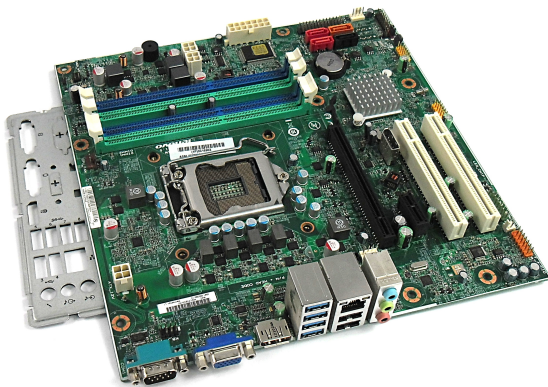


Figura: Placa madre

# Abriendo la tapa...

From Computer Desktop Encyclopedia  
Reproduced with permission.  
© 1997 Singapore Technologies

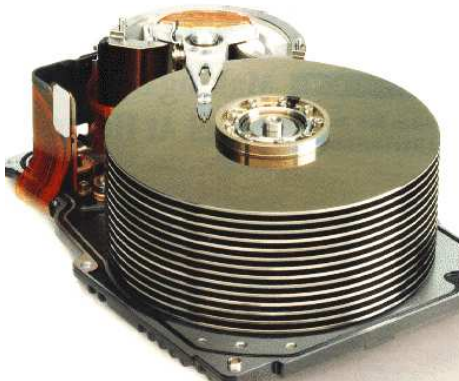


Figura: Disco magnético

# Abriendo la tapa...

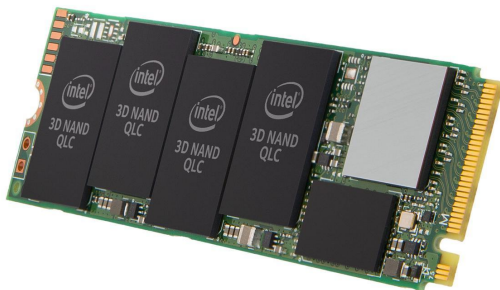


Figura: Disco de estado sólido



# Abriendo la tapa...

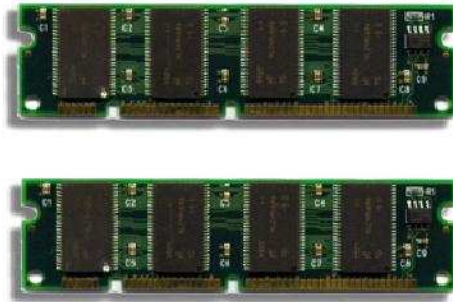


Figura: Memoria RAM

# Abriendo la tapa...

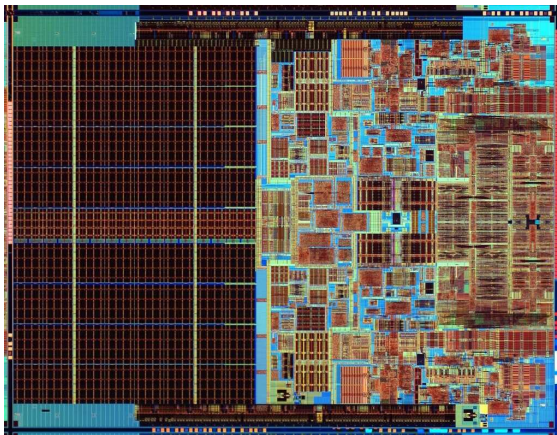


Figura: Procesador Intel Core Duo

# Estructura de la máquina de von Neumann

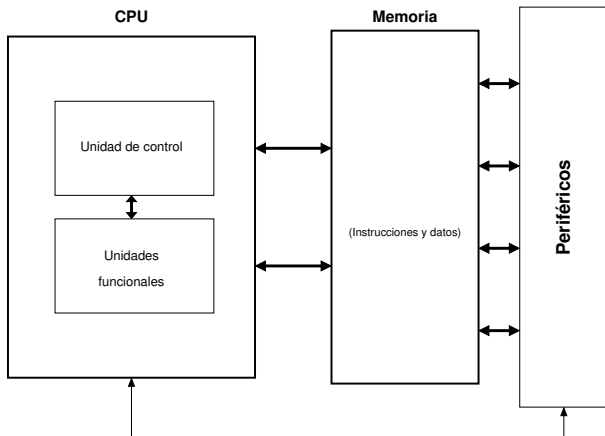


Figura: Arquitectura von Neumann

# Modelo de la memoria

**Modelo de memoria**

100	129
101	247
102	98
103	0
104	245
105	7
106	54
107	101
108	255

Dirección                      Contenido

Figura: Memoria: cada posición puede almacenar 1 byte

# Esquema de la CPU

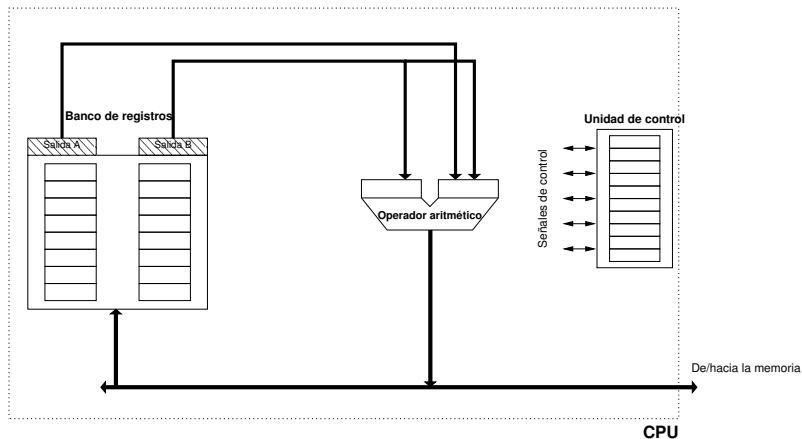


Figura: Estructura interna de la CPU

# Instrucciones

- Para manejar la máquina, hablamos su lenguaje.
  - Las “palabras” de ese lenguaje son *instrucciones*.
  - El vocabulario completo (pequeño) es el «repertorio de instrucciones».
- Las instrucciones deben ser
  - lo más *sencillas* que sea posible, pero. . .
  - deben permitir cualquier operación, es decir, el repertorio debe ser *completo*.
- Puede haber una infinidad de repertorios distintos, pero en la práctica, todos se parecen.

# Concepto de programa almacenado

- La idea clave de von Neumann es **representar** las instrucciones mediante códigos numéricos, susceptibles de ser almacenados en memoria como cualquier otro tipo de dato.
- El conjunto de todos esos códigos numéricos constituye el **lenguaje máquina**.
- Nosotros no manejamos directamente los códigos de ese lenguaje, sino un *mnemónico* asociado a cada código (= instrucción).
- Al conjunto de todos los *mnemónicos* lo llamamos *lenguaje ensamblador*.

# Ejemplos de instrucciones y registros

<b>Instrucción</b>	<b>Función</b>
ADD \$R3, \$R2, \$R1	$\$R3 \leftarrow \$R2 + \$R1$
SUB \$R3, \$R2, \$R1	$\$R3 \leftarrow \$R2 - \$R1$
ADDI \$R2, \$R1, N	$\$R2 \leftarrow \$R1 + N$
AND \$R1, \$R2, \$R3	$\$R1 \leftarrow \$R2 \& \$R3$
OR \$R1, \$R2, \$R3	$\$R1 \leftarrow \$R2   \$R3$

<b>Tipo de registro</b>	<b>Nombre</b>
De propósito general	$\$R0, \$R1, \$R2, \$R3, \dots$
Contador de programa	\$PC
Puntero de pila	\$SP



# Fases de una instrucción

- Búsqueda (*fetch*): tomar la instrucción apuntada por el registro  $\$PC$  y llevarla a la unidad de control.
- Descodificación (*decode*) de la instrucción y lectura de operandos.
- Ejecución (*execute*) de la operación.
- Almacenamiento (*write-back*) de resultados en los lugares indicados por la instrucción.

# Concepto de arquitectura

## Definición de arquitectura de un computador

Denominamos *arquitectura de un computador* al conjunto formado por el **repertorio de instrucciones** y los **registros** disponibles.

## Observación

Dos computadores que compartan la misma arquitectura pueden ejecutar exactamente el mismo programa con idéntico resultado.

# Descripción jerárquica del computador

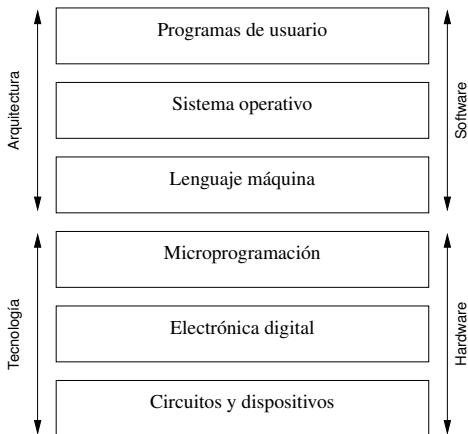


Figura: Visión lógica y jerárquica

# Lenguajes de programación

- Programar usando el lenguaje ensamblador es muy complicado.
- Se utiliza en su lugar *lenguajes de alto nivel*.
- Recuerdan, más o menos, el lenguaje natural.
- Un archivo que contiene sentencias de lenguaje de alto nivel lo denominamos *código fuente*.

# Compilador

- El compilador es un programa que convierte el código fuente en lenguaje ensamblador.

## Ejemplo en C

```
int i, j, f, g, h;  
f = (g + h) - (i + j);
```

## Lo mismo, ensamblado

```
ADD $R5, $R3, $R4  
ADD $R6, $R0, $R1  
SUB $R2, $R5, $R6
```

# Proceso de desarrollo

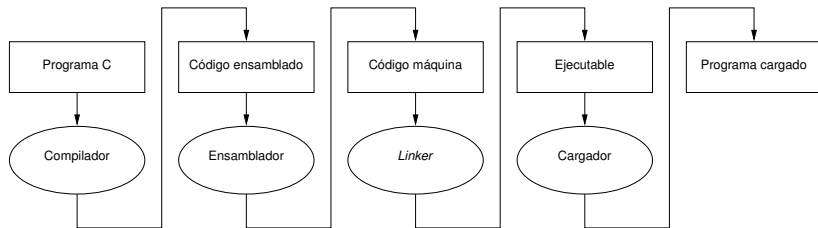


Figura: Ciclo de desarrollo y ejecución de un programa

## Un poco de historia

- Charles Babbage (Londres, 1791–1871): *analytical engine*. La primera máquina programable, con ideas tomadas de un telar que podía fabricar distintos tipos de telas mediante el uso de unas tarjetas perforadas (era un telar «programable»).
- Ada Lovelace (Londres, 1815–1852). Se la considera la primera programadora: parece que escribió un programa para calcular los números de Bernoulli<sup>1</sup> con el *analytical engine*.

---

<sup>1</sup>Los números de Bernoulli son una sucesión de racionales, conectados con varias ramas de las matemáticas. Se definen recursivamente:

$$\sum_{j=1}^m \binom{m+1}{j} B_{m-j+1} + B_0 = 0, \text{ con } m > 0 \text{ y } B_0 = 1.$$

# Estamos en la Segunda Guerra Mundial

- Proyecto ENIAC, dirigido por J. Mauchly y J.P. Eckert, durante la Segunda Guerra Mundial.
- La máquina tenía:
  - 18.000 válvulas de vacío,
  - 25 metros de larga, 2,5 metros de alta,
  - 20 registros de 10 dígitos,
  - 1.900 sumas por segundo.
  - Era programable mediante cableado y leía los datos de tarjetas perforadas.



# Proyecto ENIAC



Figura: Vista de la máquina ENIAC

# Proyecto ENIAC

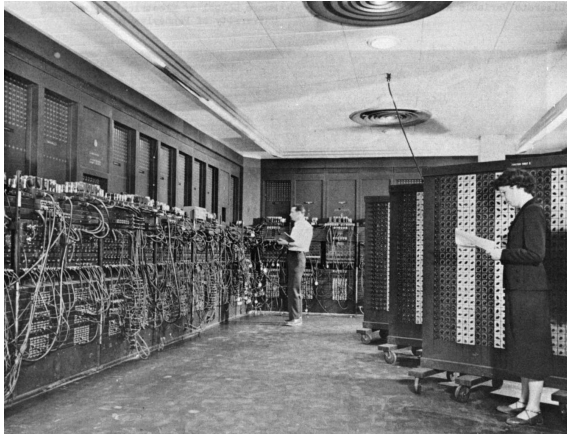


Figura: Otra vista de la máquina ENIAC

# Máquina de von Neumann

- En 1944, J. von Neumann<sup>2</sup> se unió al proyecto ENIAC y propuso codificar las instrucciones como números y almacenarlos en la memoria de la máquina.
- Con la ayuda de Goldstine y Burks, todo este trabajo cristalizó en un documento<sup>3</sup>, que se considera la fundación de los computadores tal como los entendemos ahora.

**Este es el origen de la «Arquitectura von Neumann»**

---

<sup>2</sup>John von Neumann (Budapest, 1903–Washington, 1957).

<sup>3</sup>A.W. Burks, H.H. Goldstine, J. von Neumann, *Preliminary discussion of the logical design of an electronic computing instrument*, Report to the U.S. Army Ordnance Department, 1946.

# Influencia de la tecnología

- Primera etapa:
  - Válvulas de vacío.
  - Velocidades muy lentas.
- Segunda etapa:
  - Circuitos integrados para el procesador.
  - Memorias de núcleo de ferrita (más lentas que el procesador).
  - Instrucciones complejas, para reducir su número.

# Influencia de la tecnología

- Tercera etapa
  - Aumento en la densidad de integración.
  - Aparición de memoria *cache*.
  - La complejidad de instrucciones es un inconveniente.
- Cuarta etapa
  - Mejoras de la velocidad del procesador.
  - Instrucciones sencillas, con un número total reducido al mínimo.
  - Reducción del tiempo de ciclo-máquina.
  - Ampliación de cache para contener datos e instrucciones.

## Desarrollos comerciales

- **1947:** Eckert-Mauchly Corporation. Primera máquina BINAC. No prospera.
- **1951:** E-M, adquirida por Remington-Rand. UNIVAC I. Éxito: se venden 48 máquinas a 1 millón de dólares cada una.
- **1952:** IBM 701, primer computador de IBM. Sólo se vendieron 19.
- **1964:** System/360: IBM define el concepto de *arquitectura de computadores* desarrollando la familia 360.
- **1965:** DEC PDP-8 primer mini-computador comercial. Bajo coste, *solo* 20.000 dólares.

## Desarrollos comerciales

- **1963**: Primer supercomputador: CDC 6000, creado por Seymour Cray.
- **1976**: Cray sigue liderando el desarrollo de los *grandes*: CRAY-1.
- **1977**: Primer computador personal Apple-II.
- **1981**: IBM Personal Computer (Intel y Microsoft).
- **1990**: Procesadores superescalares.
- **2000**: Multicomputadores y multinúcleos.
- **2020**: Computadores cuánticos elementales.
- **2030**: ¿Computadores cuánticos operativos?