

# Organización del Computador

07-10

## Máquina elemental

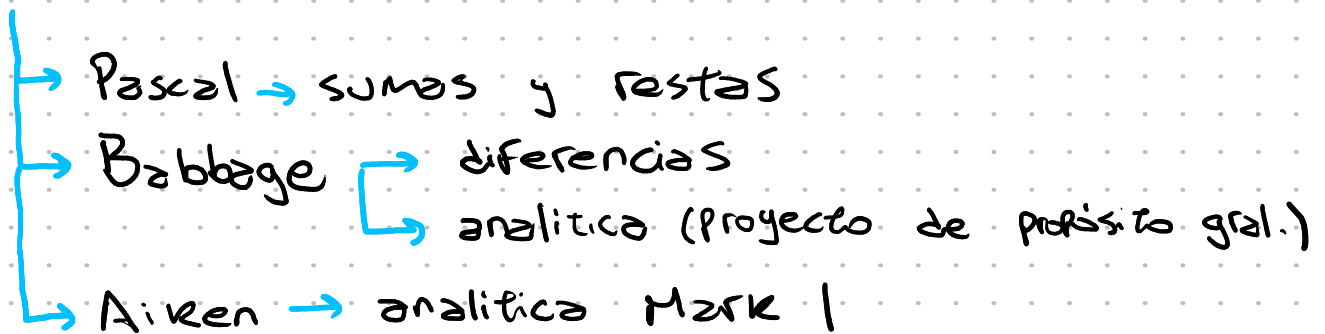
- Dato:

↓  
#

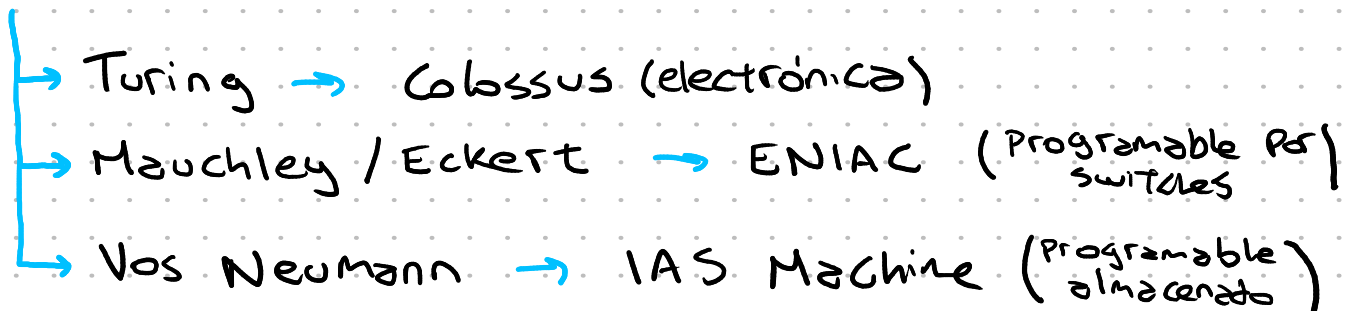
- Información:
- Bit (binary digit)
- Byte

## Clasificación de computos según generación

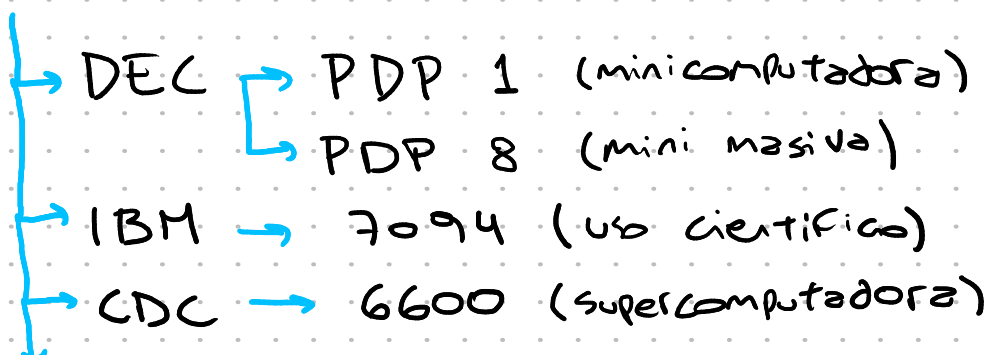
- Generación 0 - Mecánicas



- Generación 1 - Tubos de vacío



- Generación 2 - Transistores



↳ Burroughs → B5000 (alto nivel)

- Generación 3 - Circuitos Integrados

↳ DEC → PDP 11 (mini dominante)

↳ IBM → SYSTEM 360 (familia de computas)

↳ uso comercial y científico Compatibles entre si

Distintos modelos

Multiprogramación

Simulaba otras arquitecturas

- Generación 4 - Integración a muy grande escala

↳ IBM → PC (computas personales)

↳ Intel 8088

MS - DOS

↳ Apple → Apple Lisa (primera en GUI)

↳ DEC → Alpha (RISC 64 bits)

↳ COMMODORE / ATARI → hogareñas sin estándar

- Generación 5 - "Invisibles"

↳ Apple → Newton (palmtop)

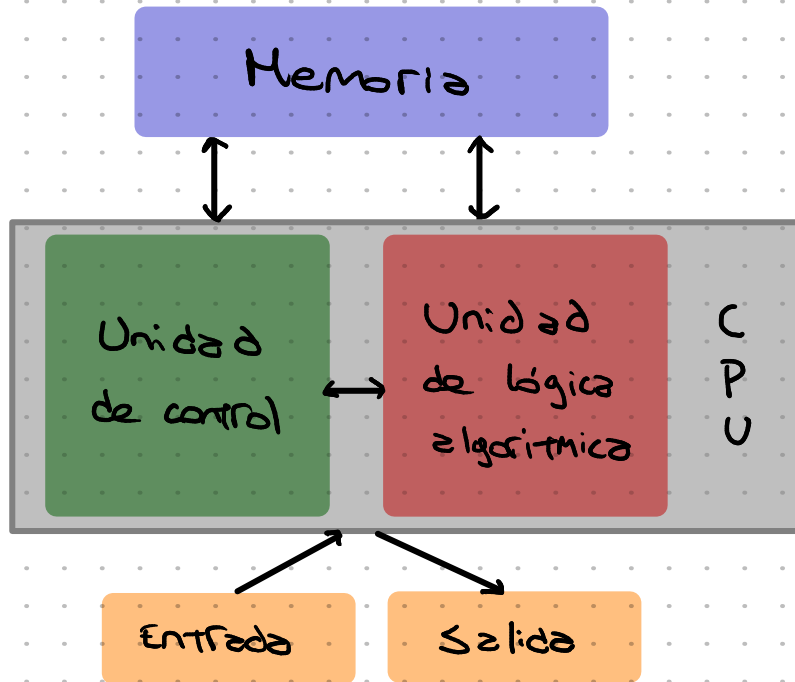
↳ Computadores → embebidos

↳ Smartwatches

Celulares

Electrodomésticos

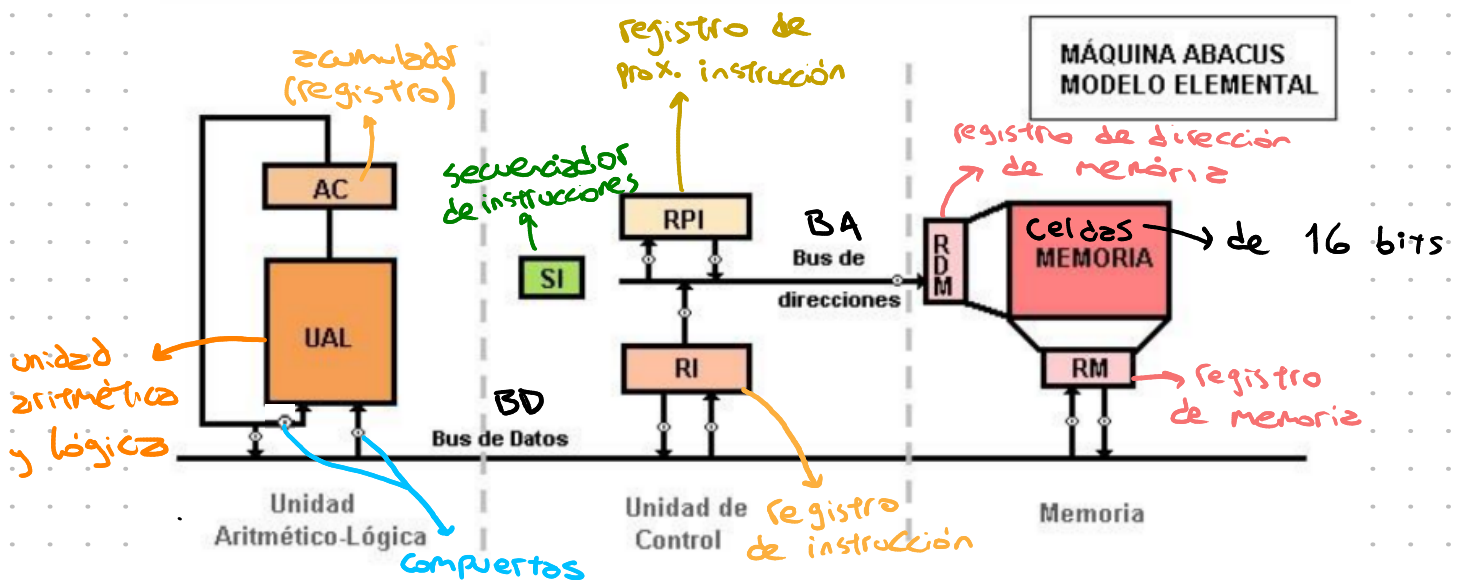
# Arquitectura de Von Neumann



Tanto las instrucciones como los datos que en ellas se usan residen en una misma memoria

Debe existir una instrucción que permita a la máquina no seguir con la secuencia de ejecución

## Máquina Abacus



Los buses no almacenan información, la transportan. Las compuertas permiten (o no) el paso de esa info y tienen un único sentido.

## Características

una dirección (operando)

↳ No existe el  $A+B$ , tendríamos que guardar A en el acumulador y después usar el operando + B

acumulador

memoria dividida en celdas

## Operaciones

Memoria

Lectura

Escritura

UAL

Aritméticas (sumar, restar, multiplicar, etc.)

Lógicas (not, and, or, etc.)

Comparación / Bifurcación

Transferencia (cargar, almacenar, etc.)

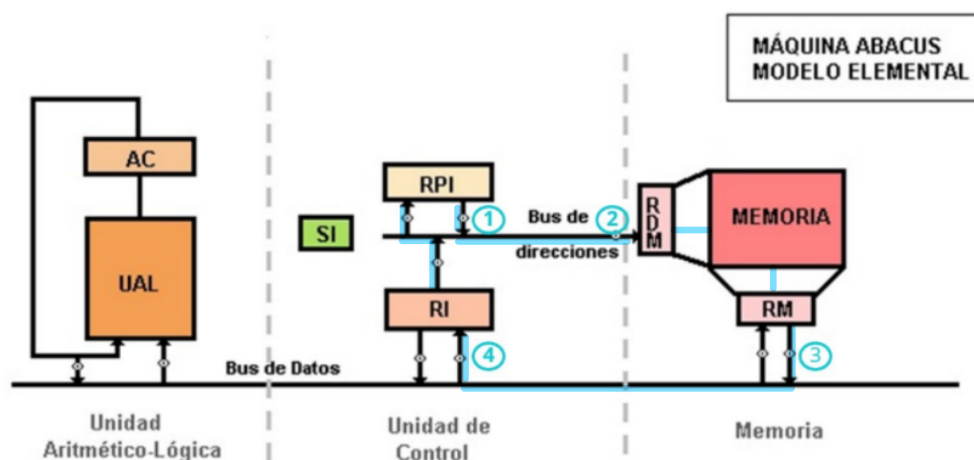
## Instrucciones

Cada instrucción tiene 2 fases

### 1) Fase de búsqueda (Fetch)

Es universal para toda instrucción dentro de la misma máquina

Traslada los bits que representan la instrucción al RI



1. El SI digita la apertura de las compuertas 1 y 2
2. Se copian los datos al bus de direcciones
3. Se transporta la información al RDM (registro de dirección de memoria)
4. El SI digita el cierre de las compuertas 1 y 2
5. Se envía la orden de lectura a la memoria con la dirección de memoria de RDM
6. Se copian los bytes que representan la instrucción (que estaba almacenado en la celda de memoria) al RM
7. El SI digita la apertura de las compuertas 3 y 4
8. Se transporta la instrucción al RI
9. El SI digita el cierre de las compuertas 3 y 4
10. El SI le emite una señal al RPI para que haga el autoincremento (y se actualice la instrucción)

La forma de escribir todo lo anterior es

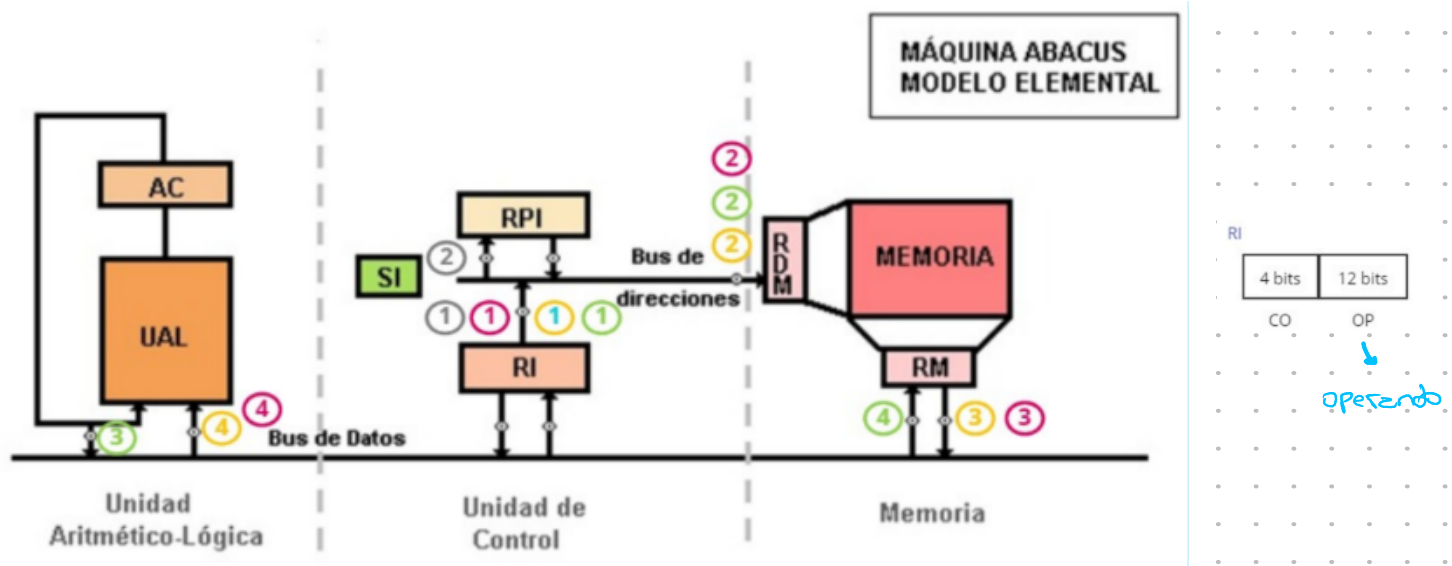
Microinstrucción


{ (RPI) ----> RDM  
 ((RDM)) ----> RM      ← Operación de lectura  
 (RM) ----> RI  
 (RPI)+1 ----> RPI      ← Incremento vía SI

Los Paréntesis representan "el contenido de"


## 2) Fase de ejecución

Las ejecuciones son particulares para cada instrucción aún dentro de la misma máquina




**Cargar**  Ej. (RI) = 1200 [16]


(OP) ----> RDM  
( (RDM) ) ----> RM ← Operación de lectura  
(RM) ----> AC

**Almacenar**  Ej. (RI) = 220F [16]

(OP) ----> RDM  
(AC) ----> RM  
(RM) ----> (RDM) ← Operación de escritura

**Sumar**  Ej. (RI) = 320E [16]

(OP) ----> RDM  
( (RDM) ) ----> RM ← Operación de lectura  
(RM) + (AC) ----> AC

**Bifurcar**  Ej. (RI) = 721A [16]

(OP) ----> RPI ← Si se cumple la condición

## Unidad de control

### Señales de gobierno

- De transferencia (ej (OP) → BA, (RPI) + 1 → RPI)
- De gobierno de memoria (lectura, escritura)
- De gobierno de la UAL (ej ejecutar una suma, resta, etc.)