

**CONQUER**  
**BLOCKS**

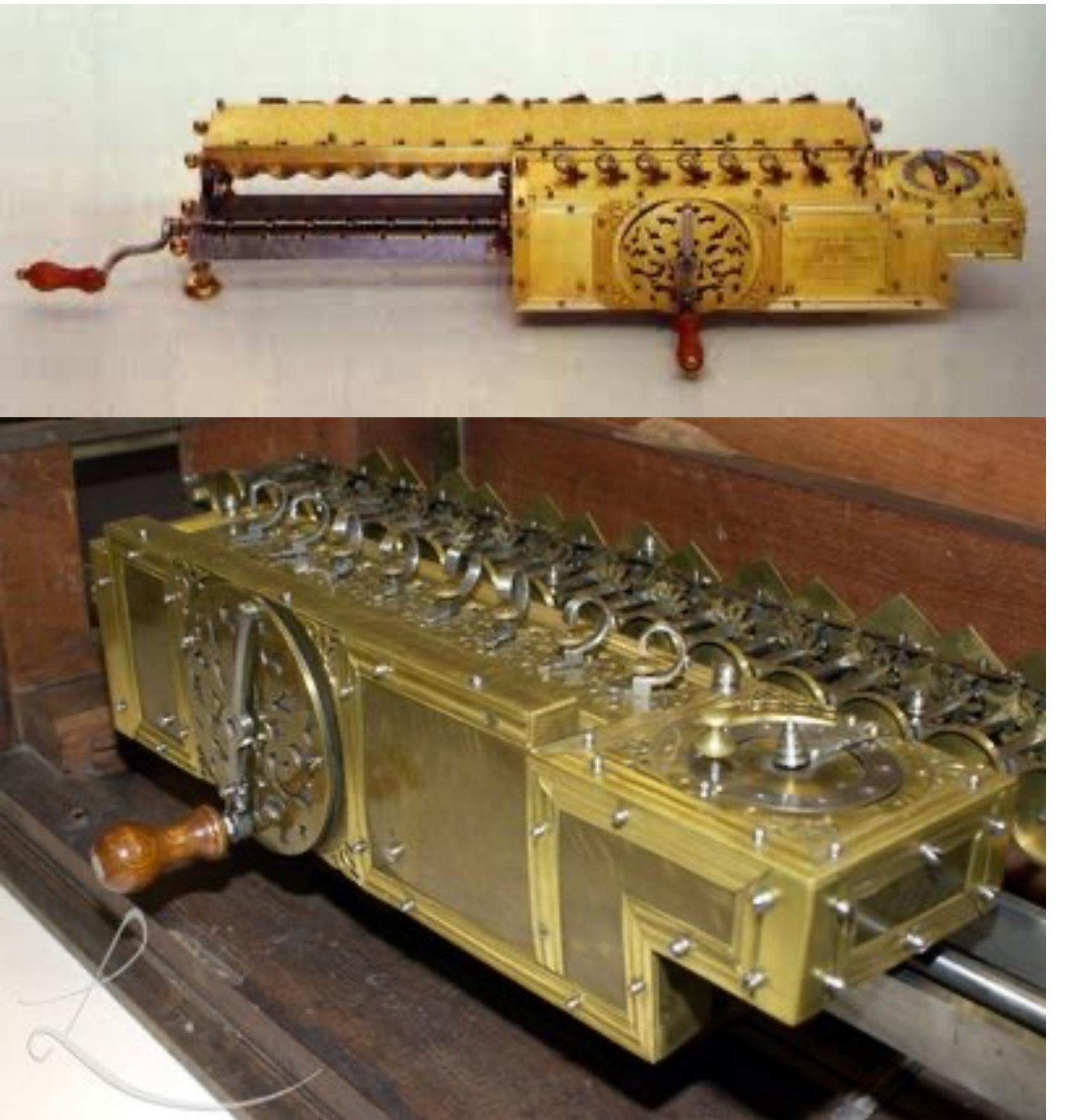
# BASES DE LA INFORMÁTICA

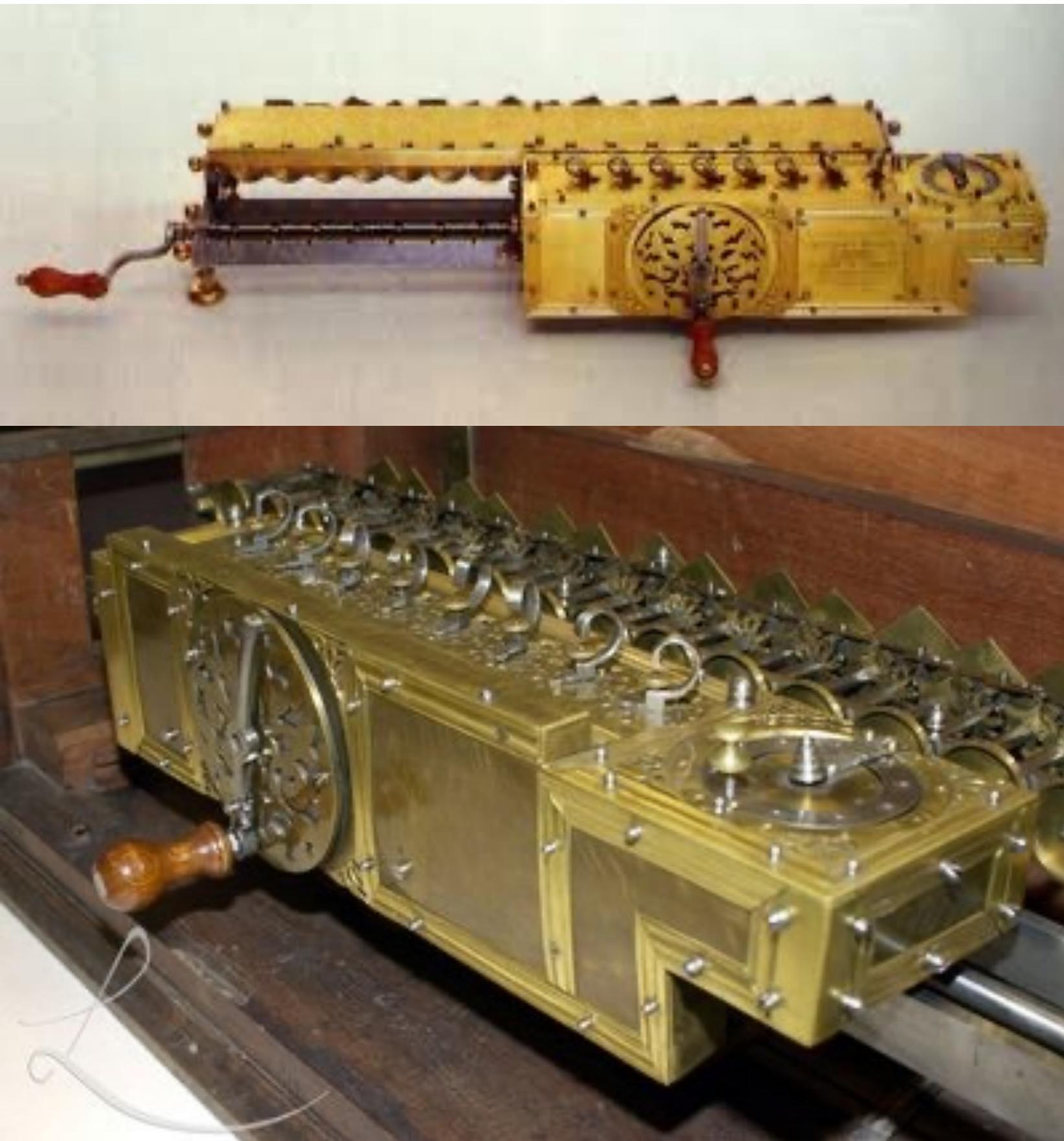
ESTRUCTURA GENERAL

---

**CONQUER** BLOCKS



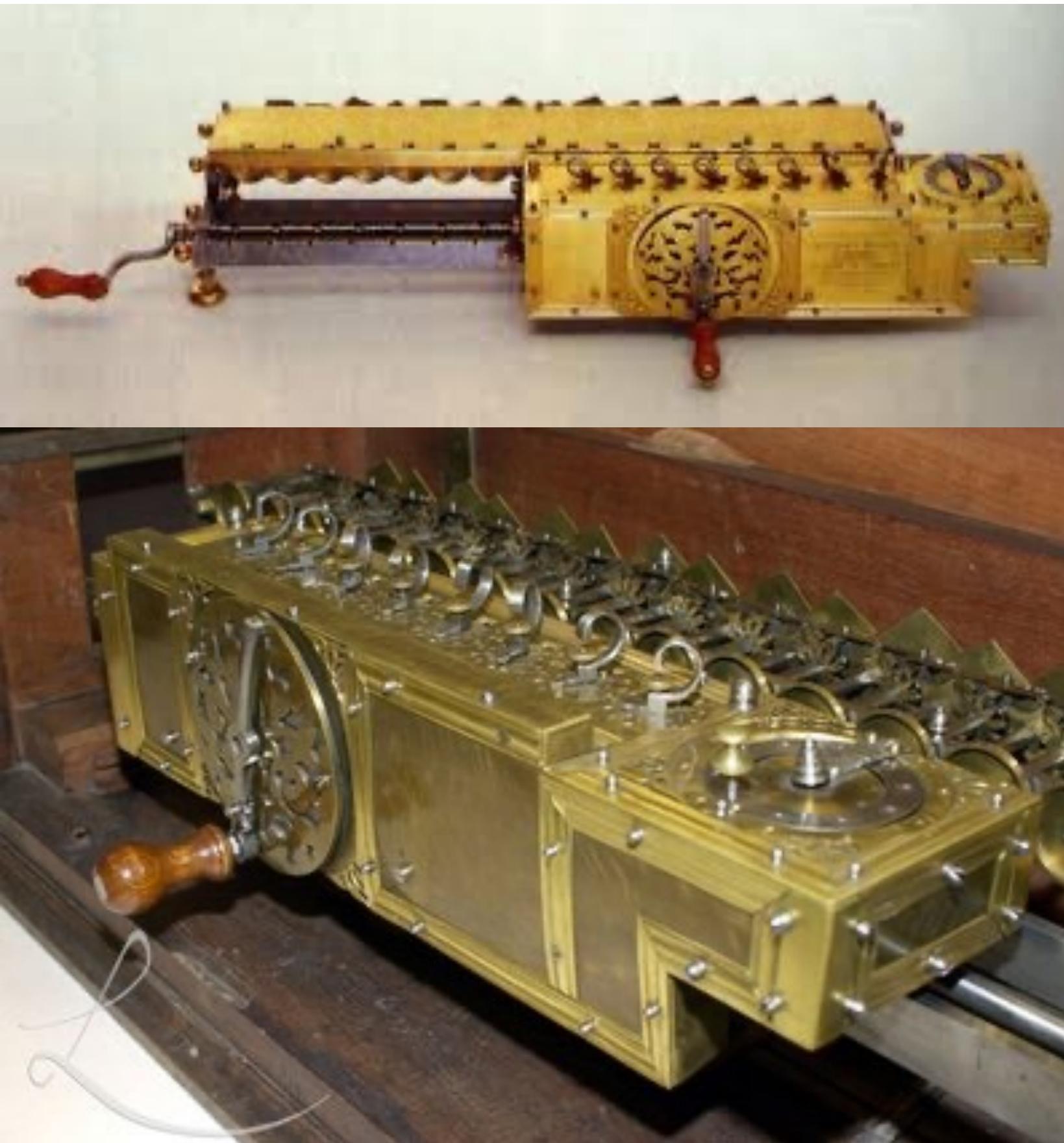




Gottfried von Leibniz

1671



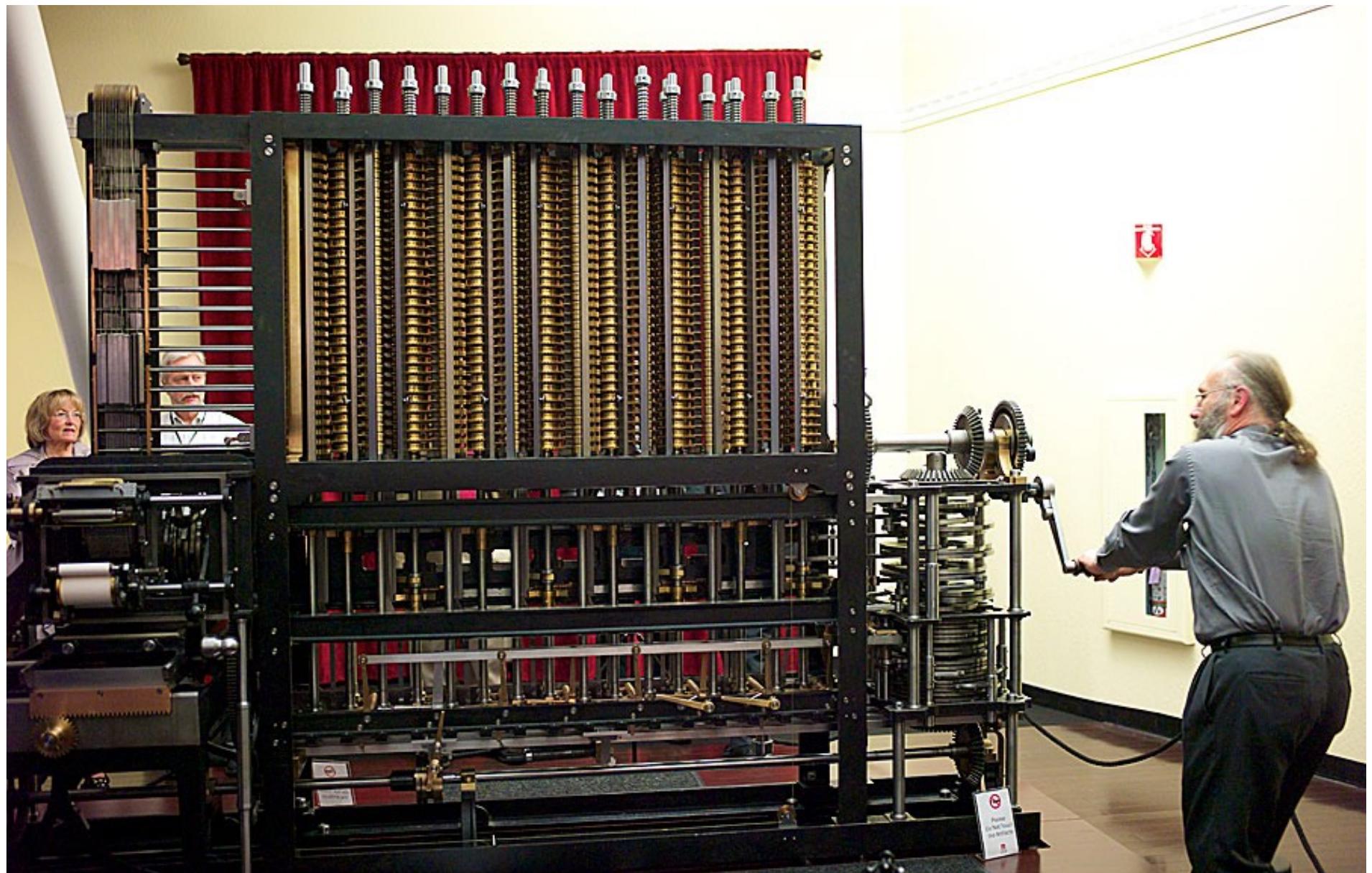


## Gottfried von Leibniz

1671

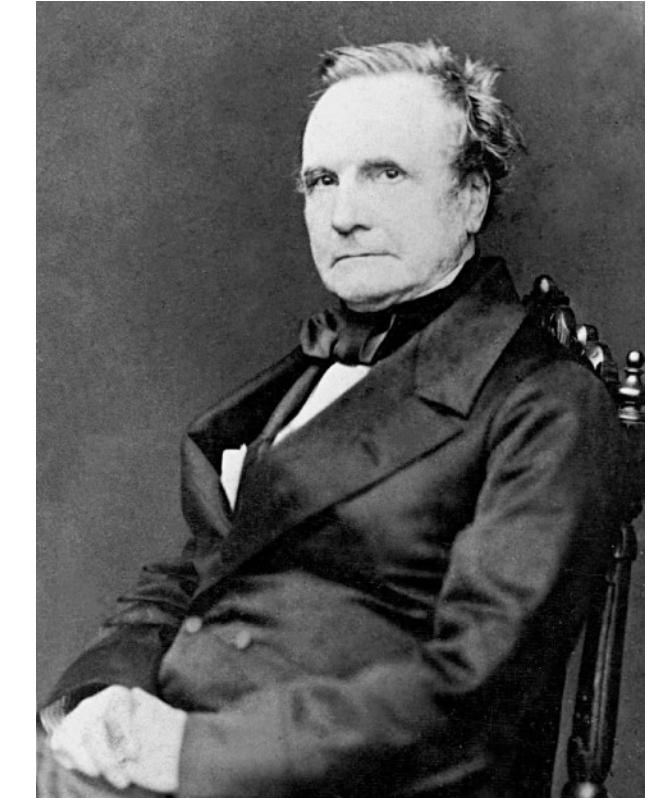


“Es indigno para hombres excelentes perder horas como esclavos en el trabajo del cálculo, cuando este podría relegarse con certeza a cualquier otra persona si se utilizaran máquinas.”



Charles Babbage

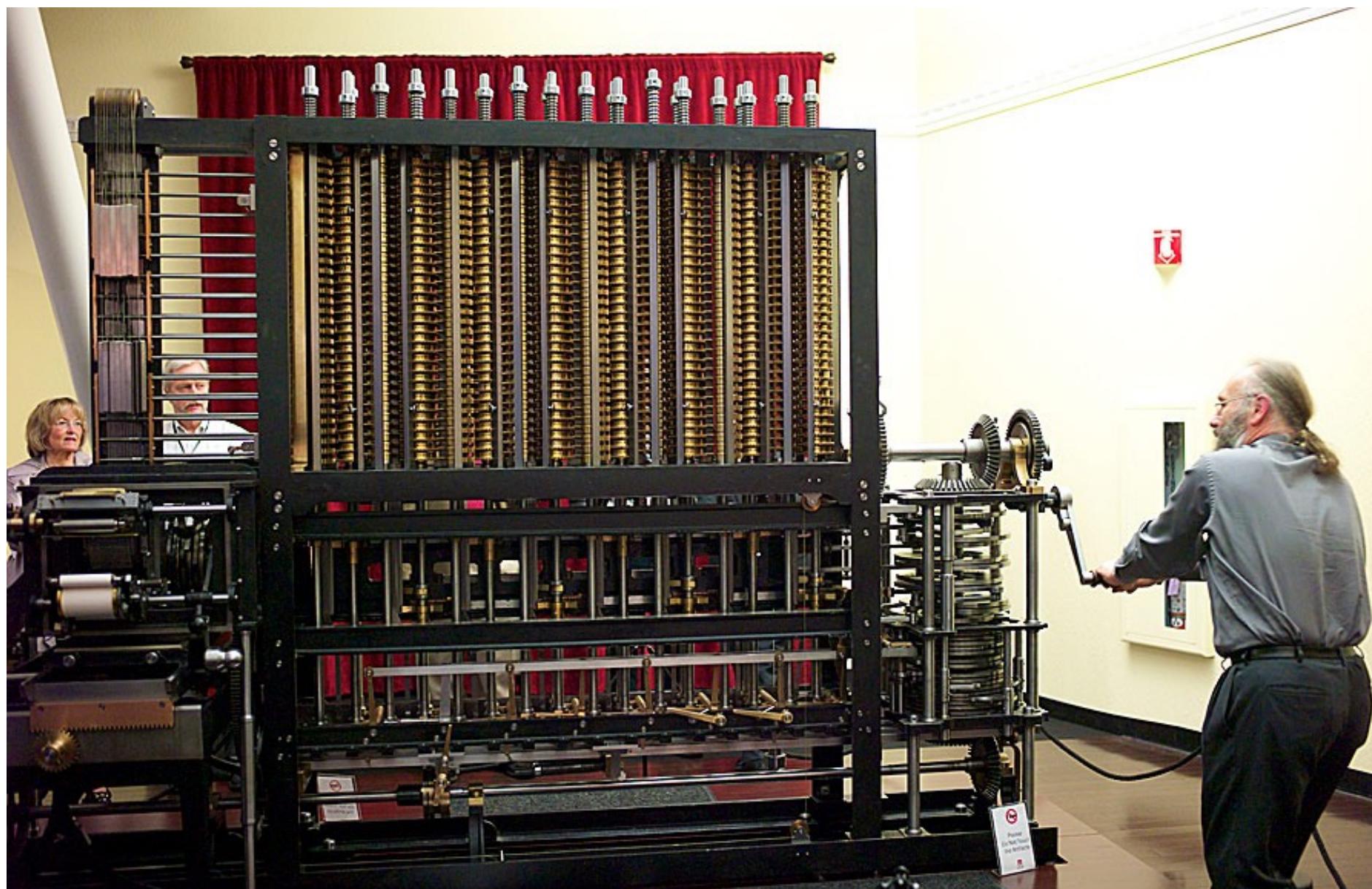
1822



Máquina diferencial o analítica



## Lady Ada Byron



1822

Diagram belonging to Note D

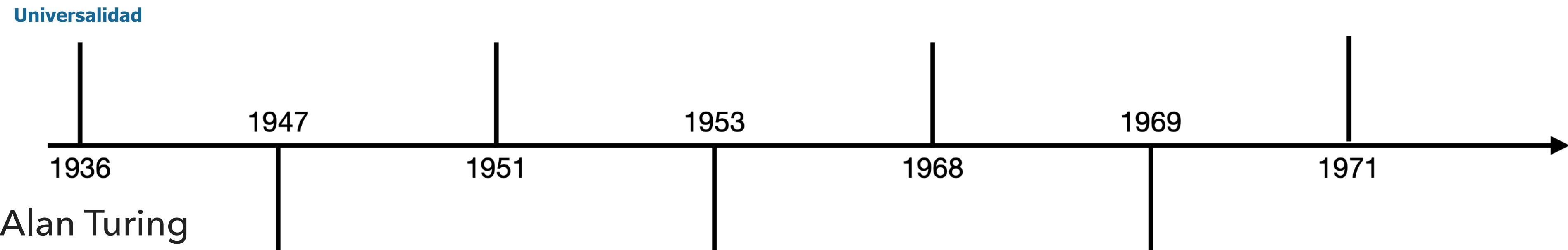
Number of Operations Name of Operations	Variables for Data										Working Variables			
	IV <sub>0</sub>	IV <sub>1</sub>	IV <sub>2</sub>	IV <sub>3</sub>	IV <sub>4</sub>	IV <sub>5</sub>	IV <sub>6</sub>	IV <sub>7</sub>	IV <sub>8</sub>	IV <sub>9</sub>	IV <sub>10</sub>	IV <sub>11</sub>	IV <sub>12</sub>	IV <sub>13</sub>
1 × m	m	n	d	m'	n'	d'								
2 × .....		n		m'										
3 × .....			d											
4 × .....			0				d'							
5 × 0							0							
6 × .....			0	0										
7 + .....							0	0						
8 - .....								0	0					
9 - .....									0	0				
10 ÷ .....														
11 ÷ .....														

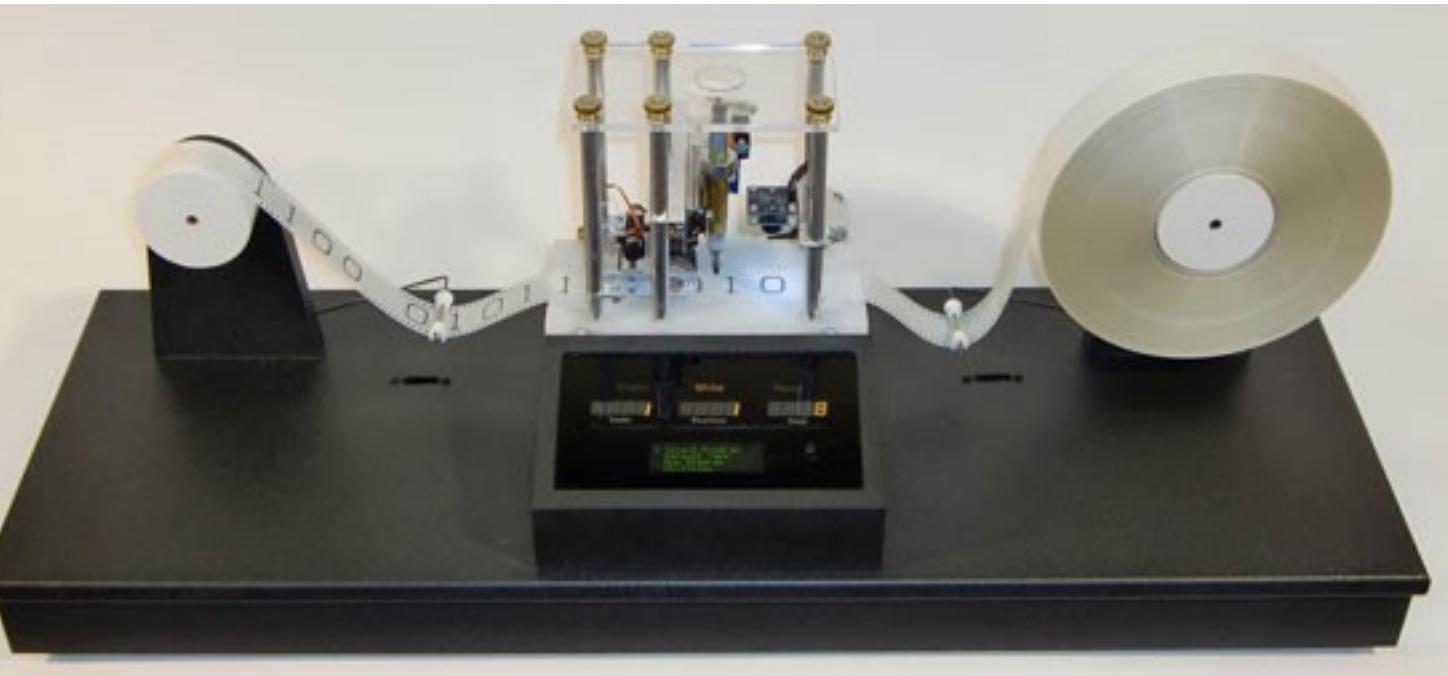
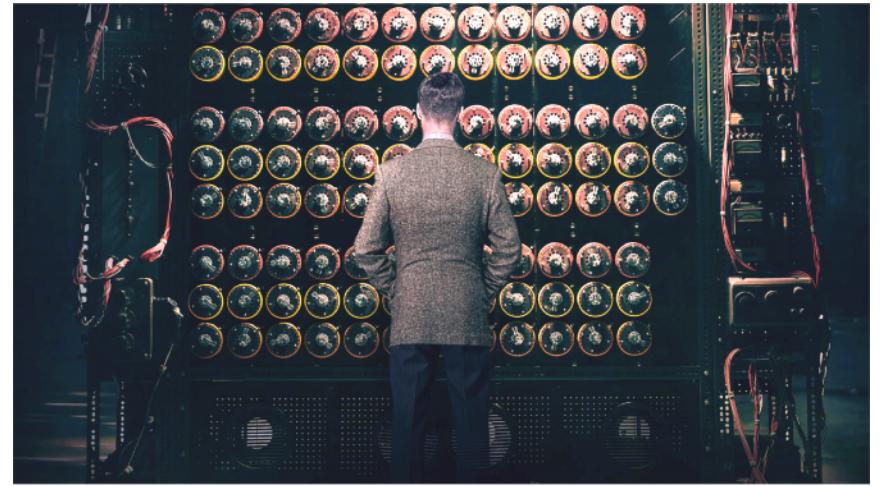
$\frac{d'm - dn'}{mn' - m'n} = y$

Máquina diferencial o analítica

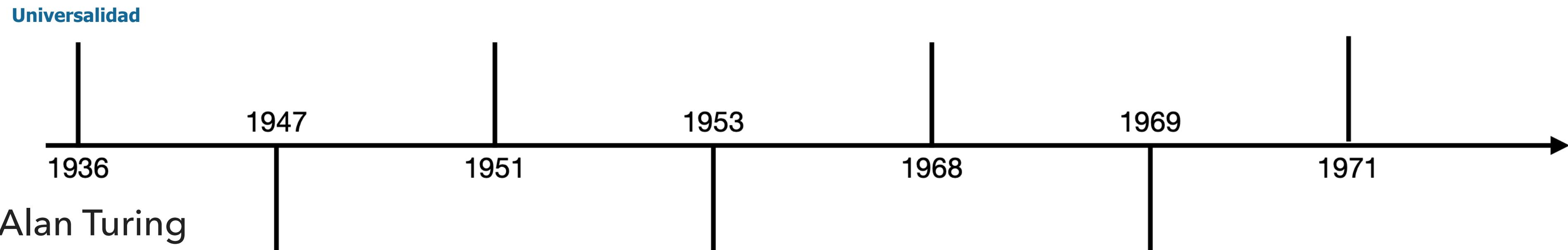


Computabilidad como procesamiento de símbolos



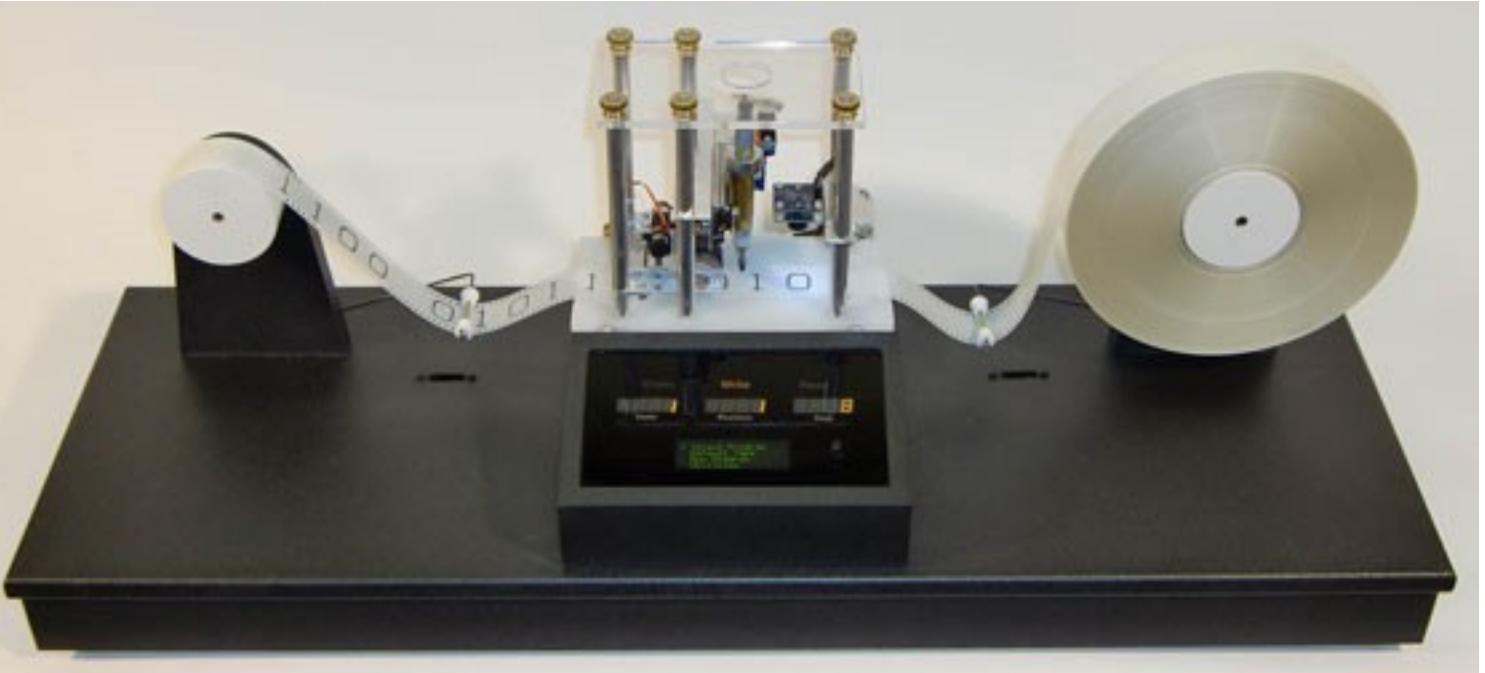


Computabilidad como procesamiento de símbolos



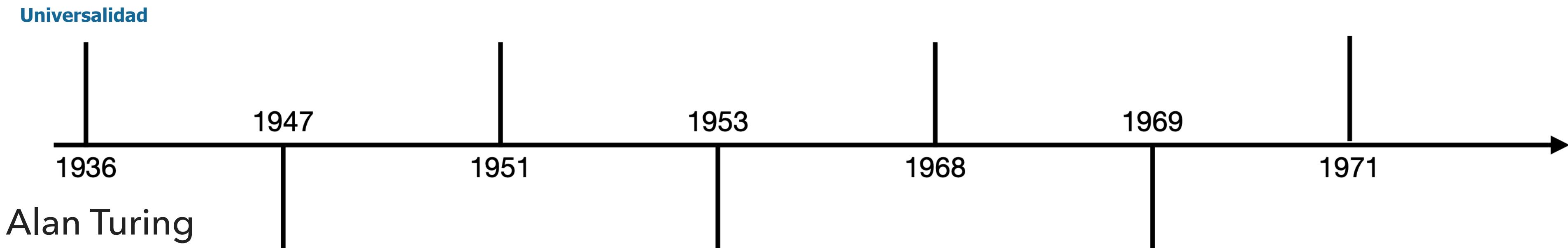


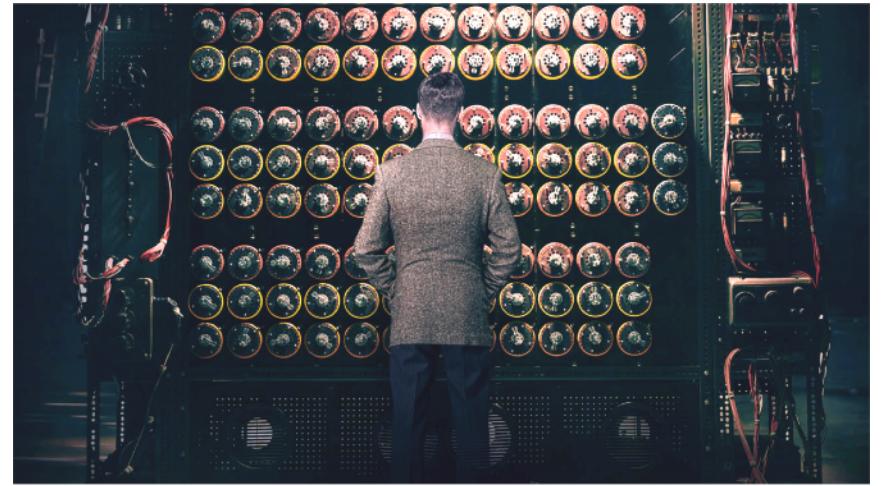
Computabilidad como procesamiento de símbolos



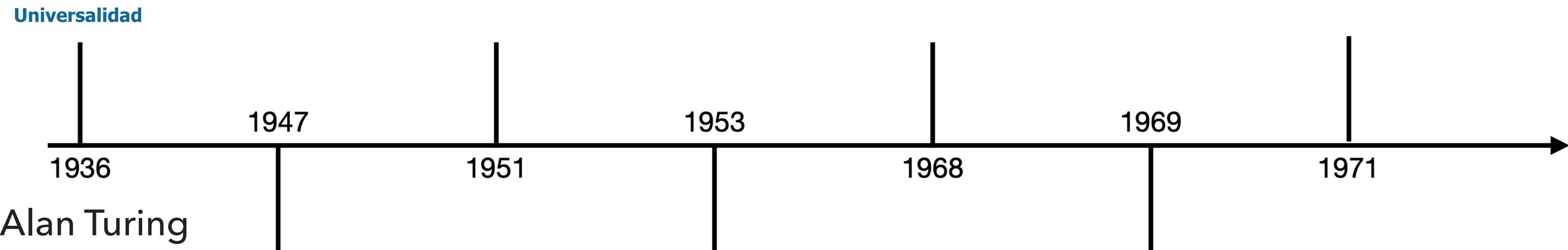
Se puede calcular **cualquier cosa** usando 6 pilares o ‘primitivos’:

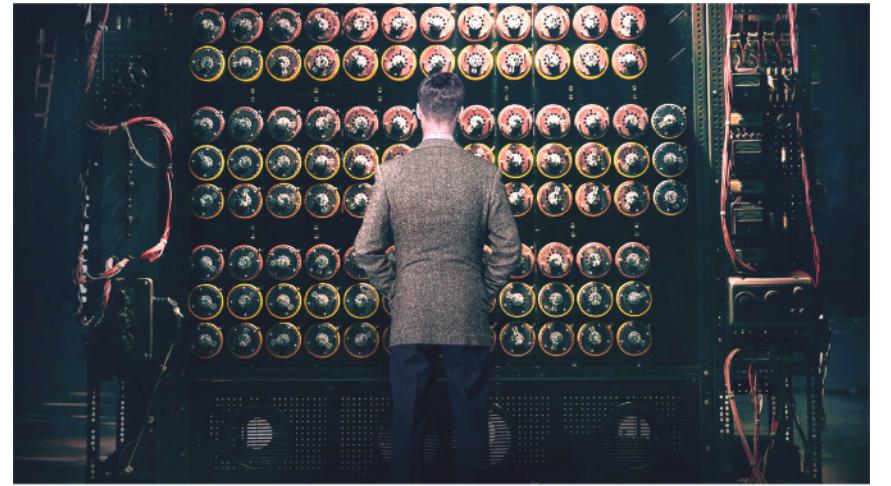
move a la derecha/izquierda  
lee / escribe  
scanea  
no hagas nada



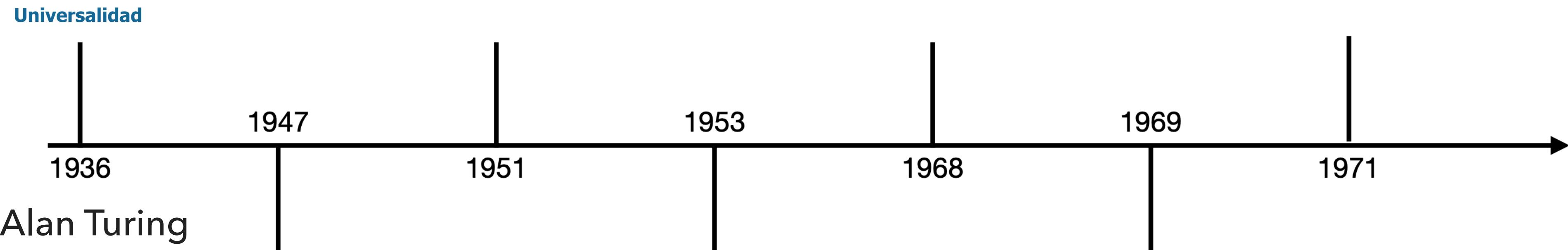


Computabilidad como procesamiento de símbolos





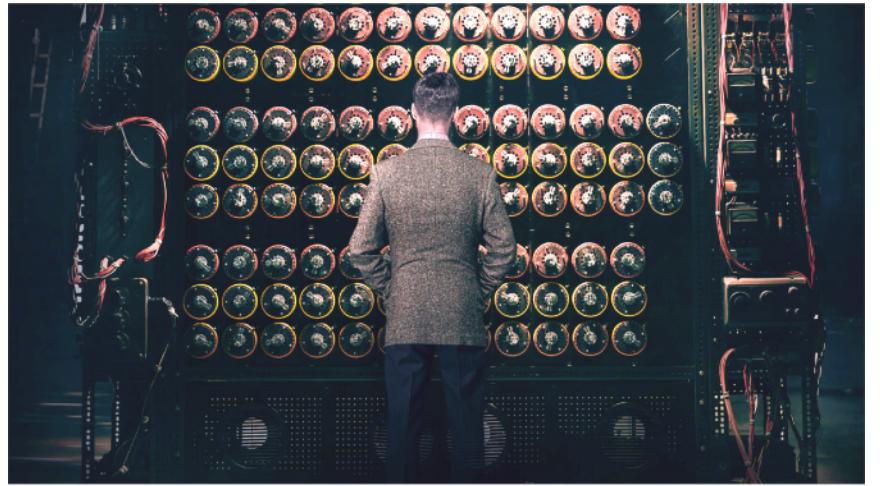
Computabilidad como procesamiento de símbolos



Transistores



## UNIVAC I



Computabilidad como procesamiento de símbolos

Universalidad

1936  
1947

Alan Turing

1era generación  
Se vendieron 46 copias

1951  
1953  
1968

1969  
1971



Transistores



## UNIVAC I



Computabilidad como procesamiento de símbolos

Universalidad

1936  
1947

Alan Turing

1era generación  
Se vendieron 46 copias

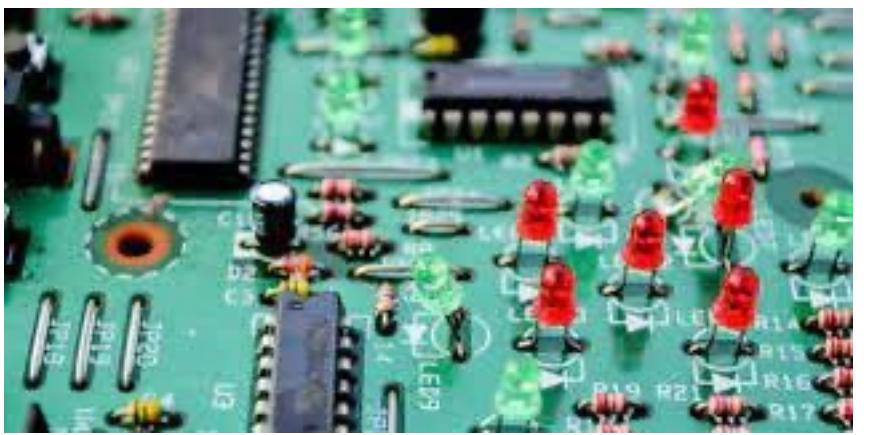
1951

1953  
1968

1969

1971

1er lenguaje de programación



Transistores



FORmula TRANslator



## UNIVAC I



Computabilidad como procesamiento de símbolos

Universalidad



1era generación  
Se vendieron 46 copias



Interfaz gráfica

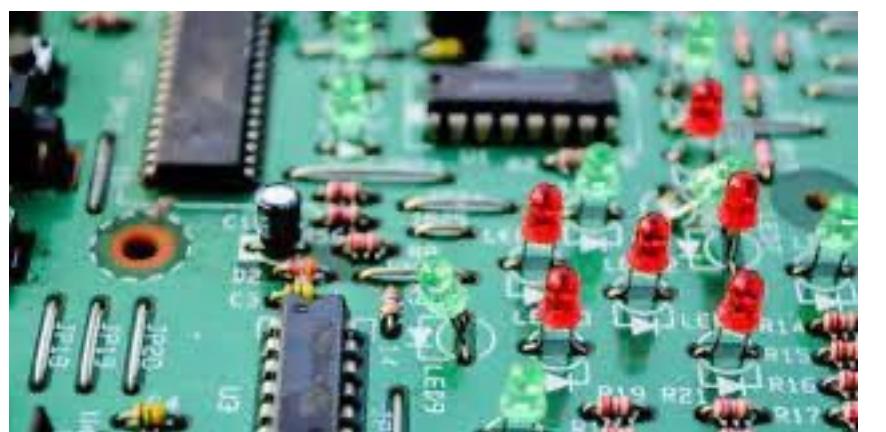
1936  
1947

1951  
1953

1968  
1969

1971

Alan Turing



Transistores



FORmula TRANslator



## UNIVAC I



Computabilidad como procesamiento de símbolos

Universalidad

1936  
1947

Alan Turing



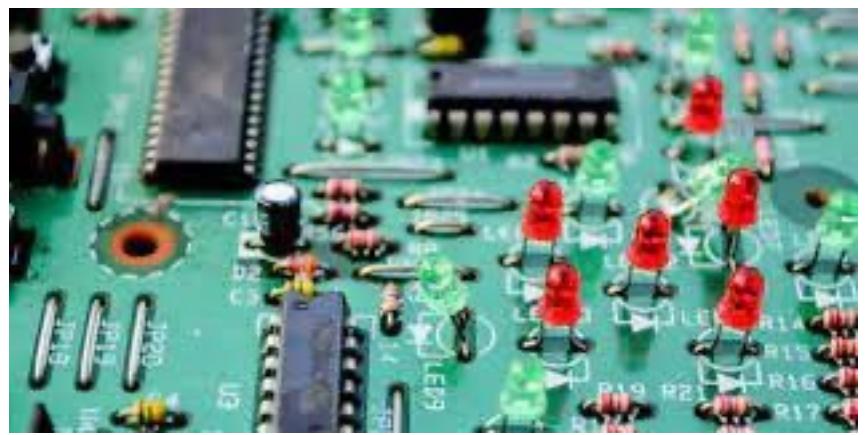
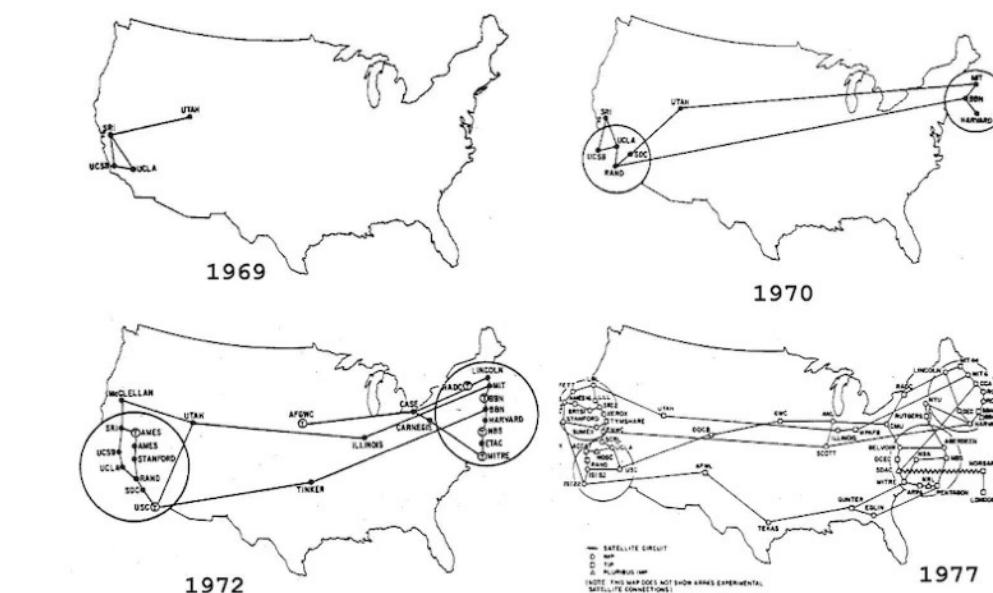
1era generación  
Se vendieron 46 copias



Interfaz gráfica

1969  
1971

1era red de comunicación



Transistores



FORmula TRANslator



## UNIVAC I



Computabilidad como procesamiento de símbolos

Universalidad



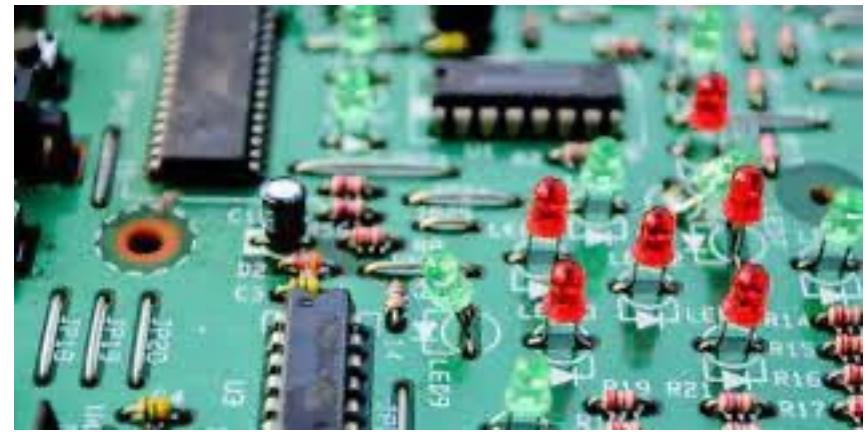
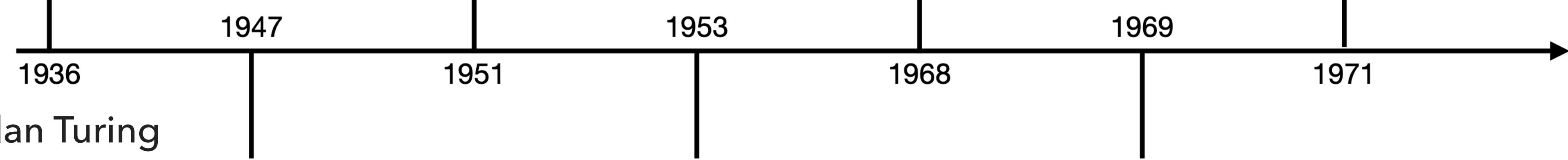
1era generación  
Se vendieron 46 copias



Interfaz gráfica



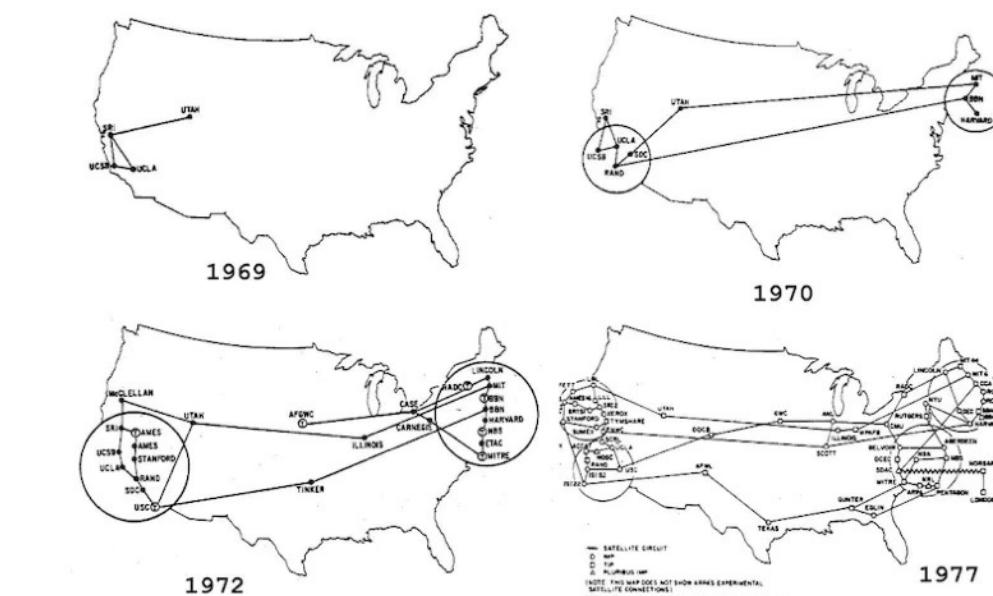
Transmisión de información (código)  
entre ordenadores



Transistores



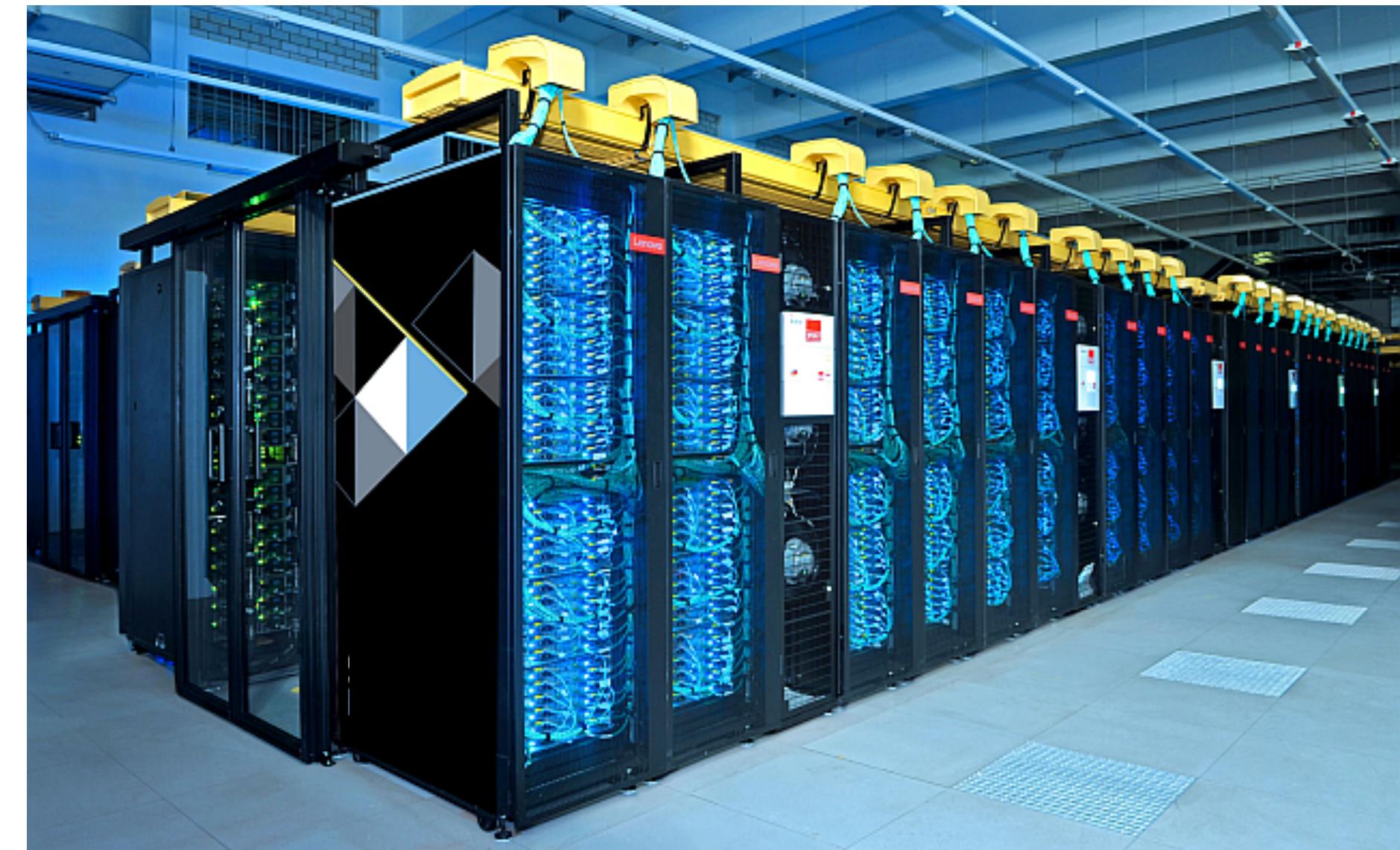
FORmula TRANslator



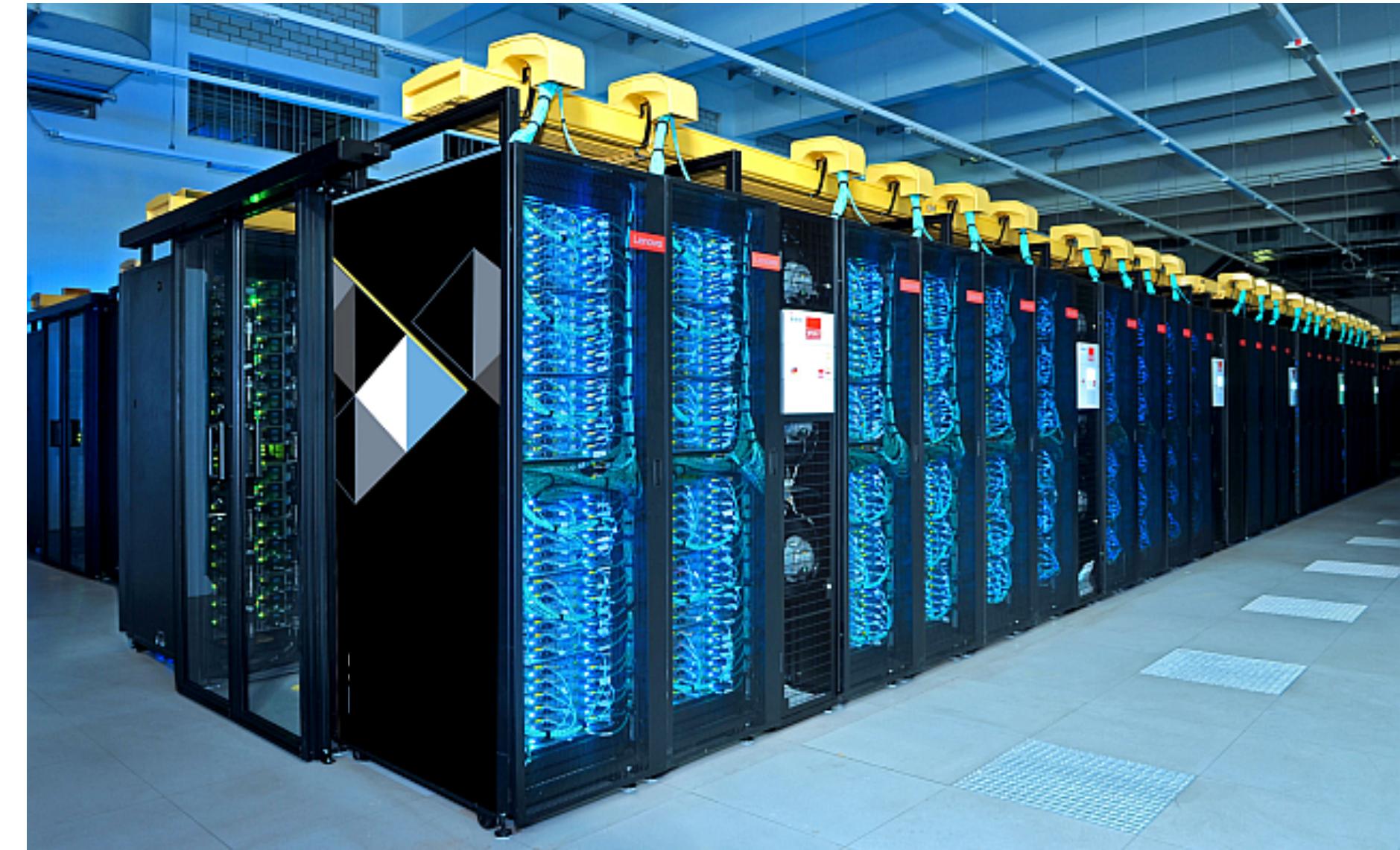
**CONQUER** BLOCKS



**CONQUER** BLOCKS



**CONQUER** BLOCKS



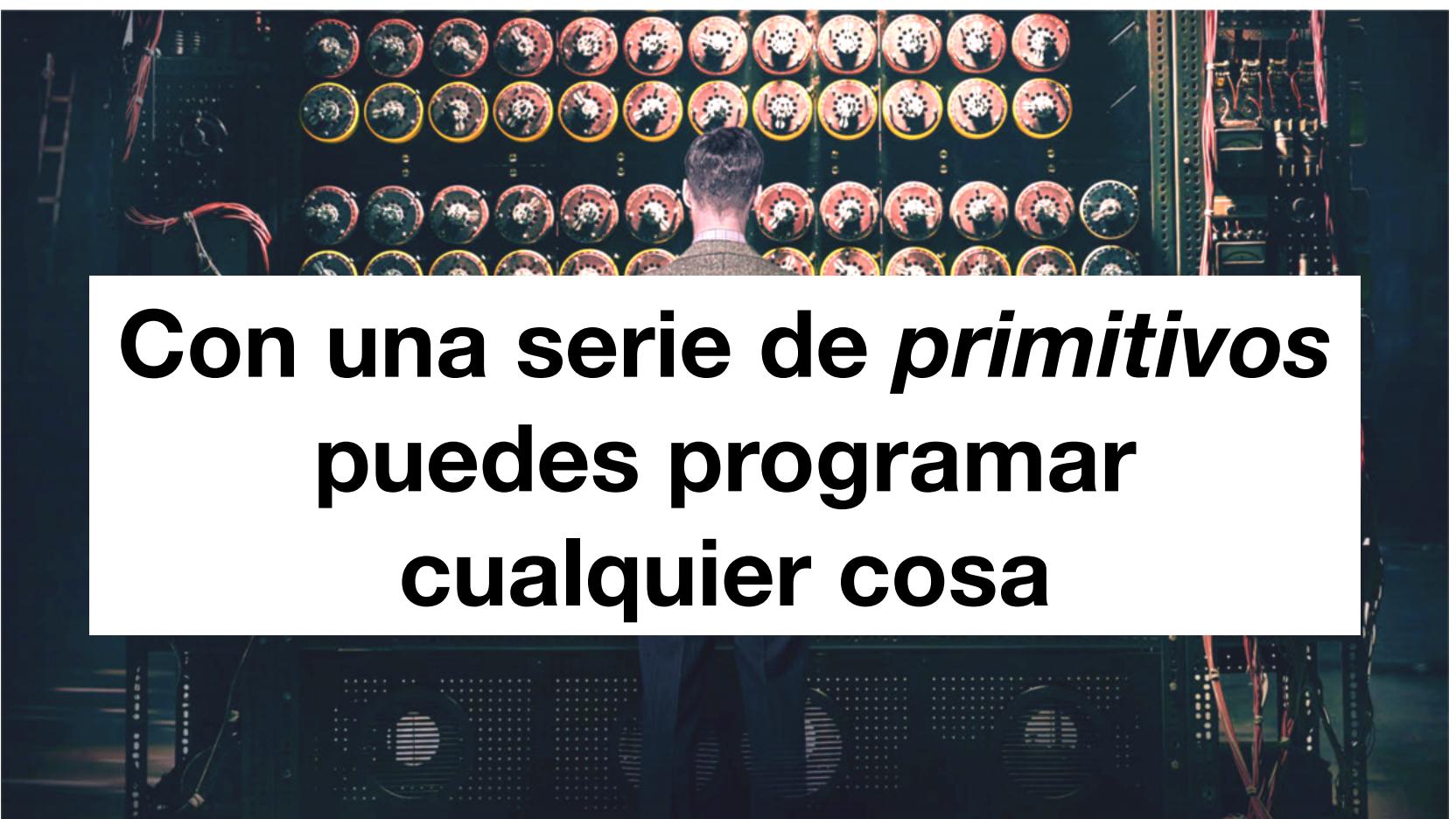
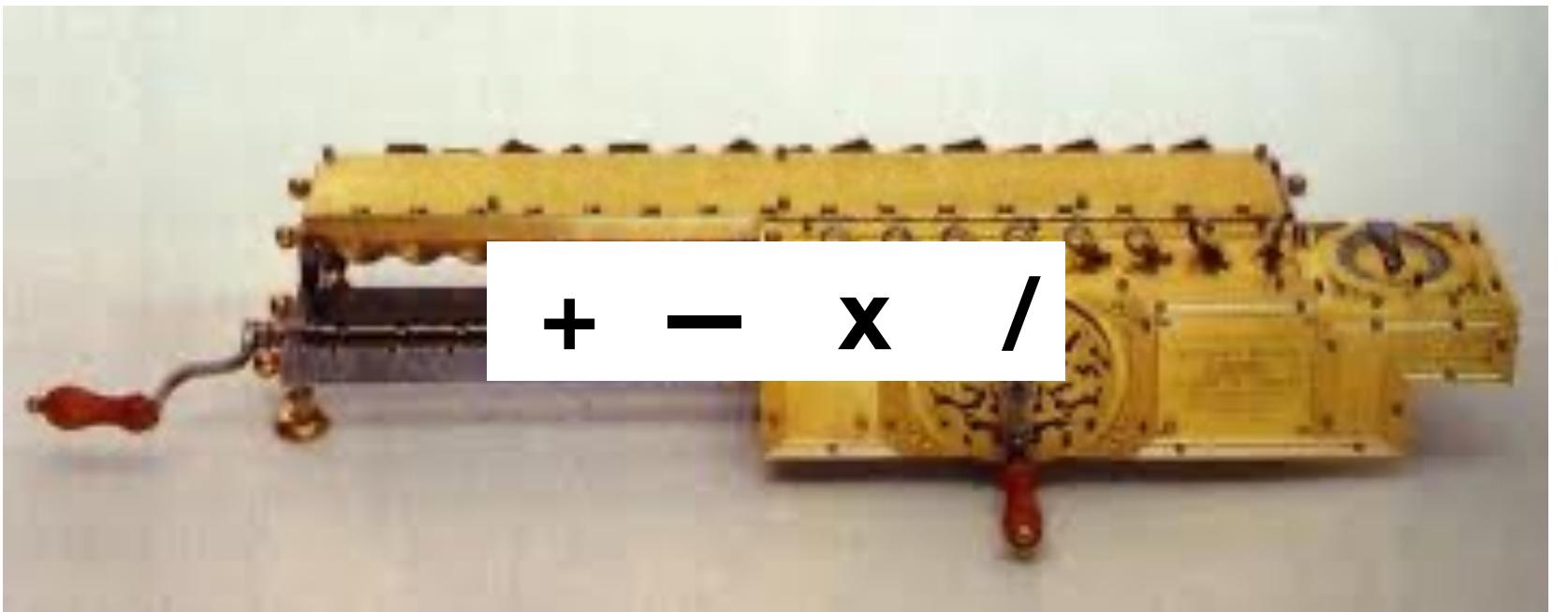


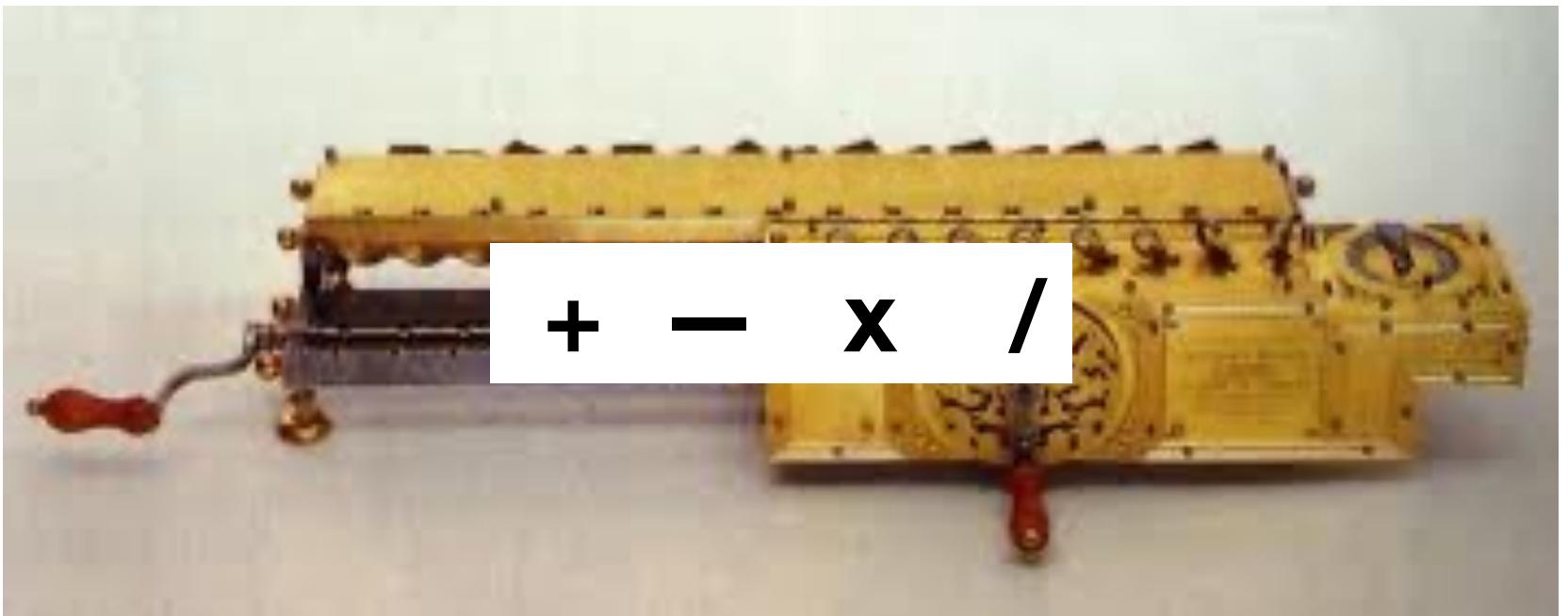


¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?

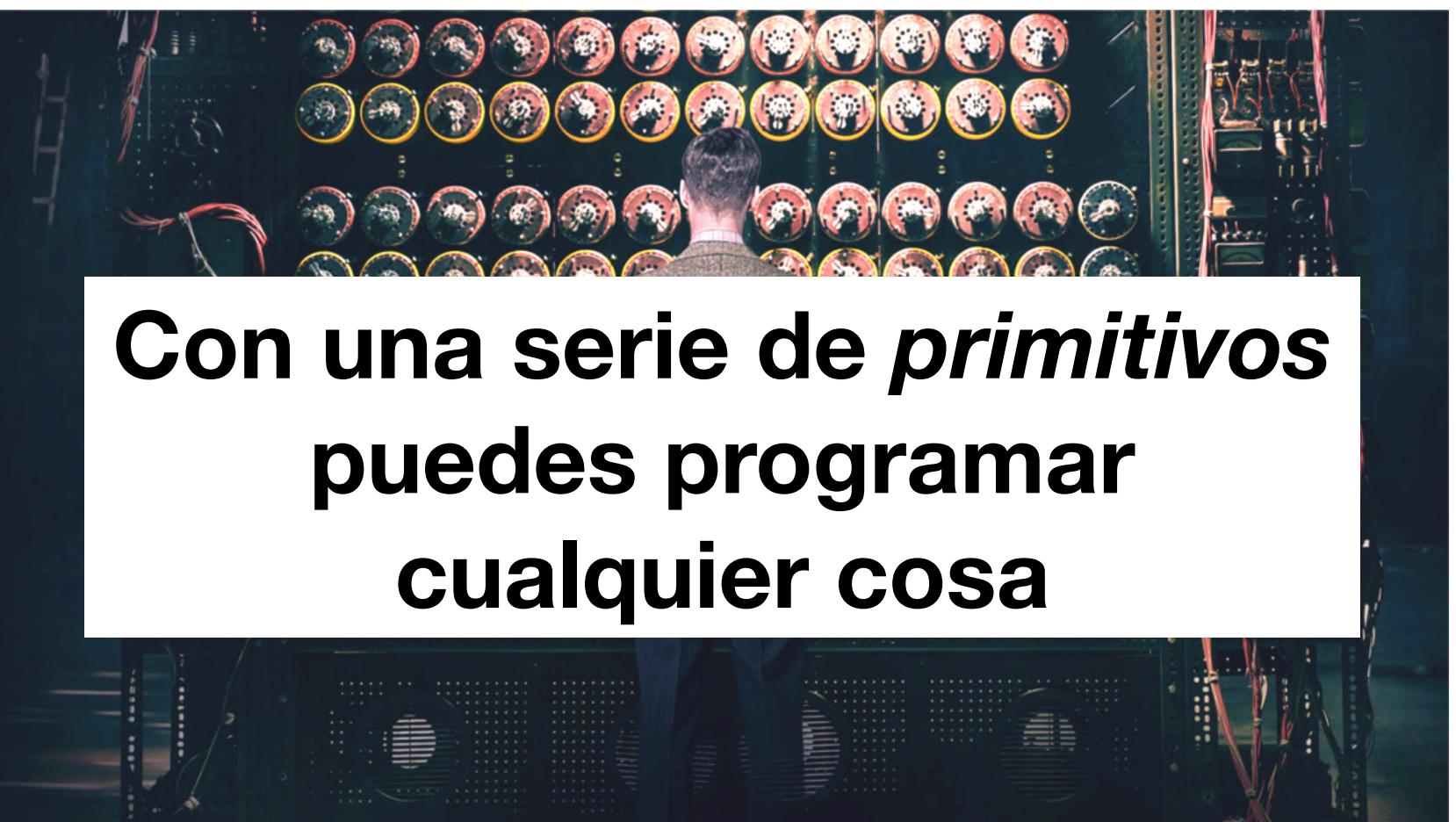
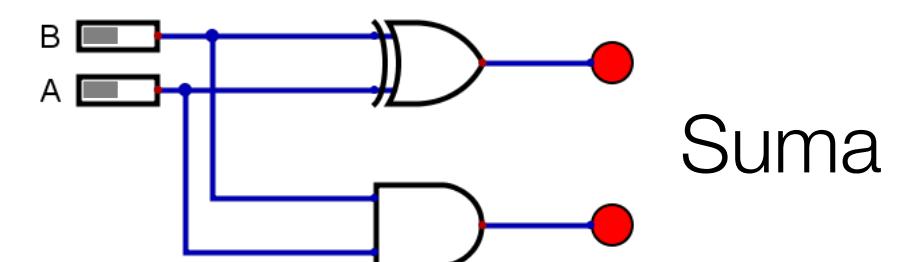


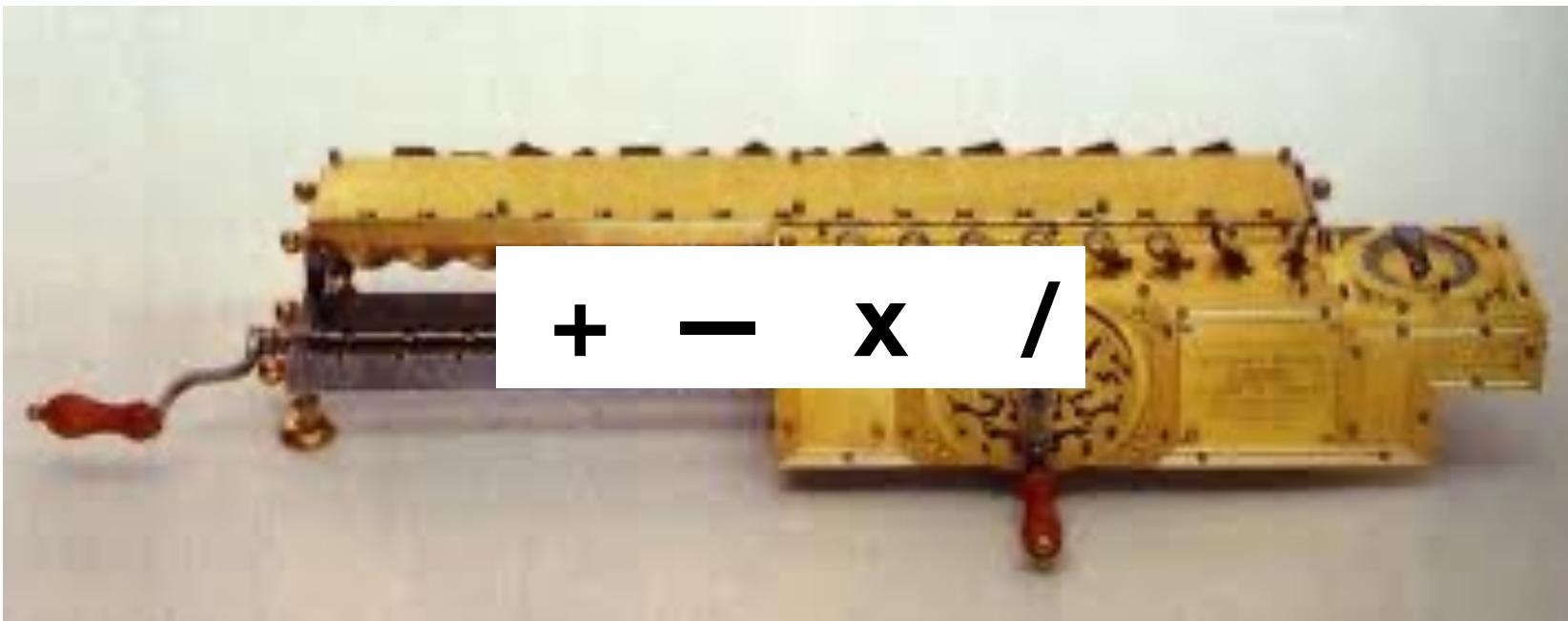
¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?



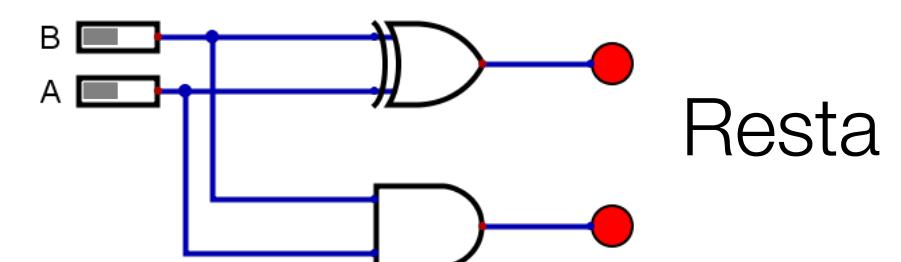
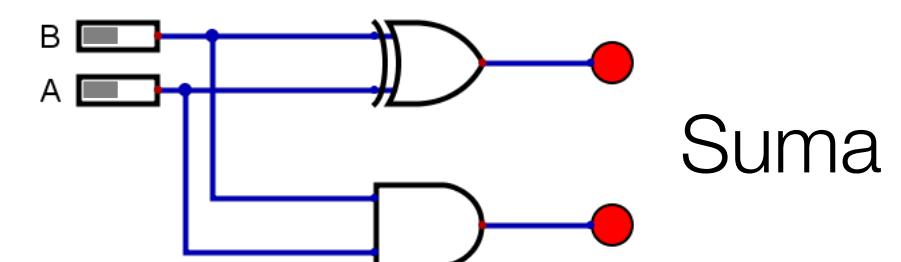


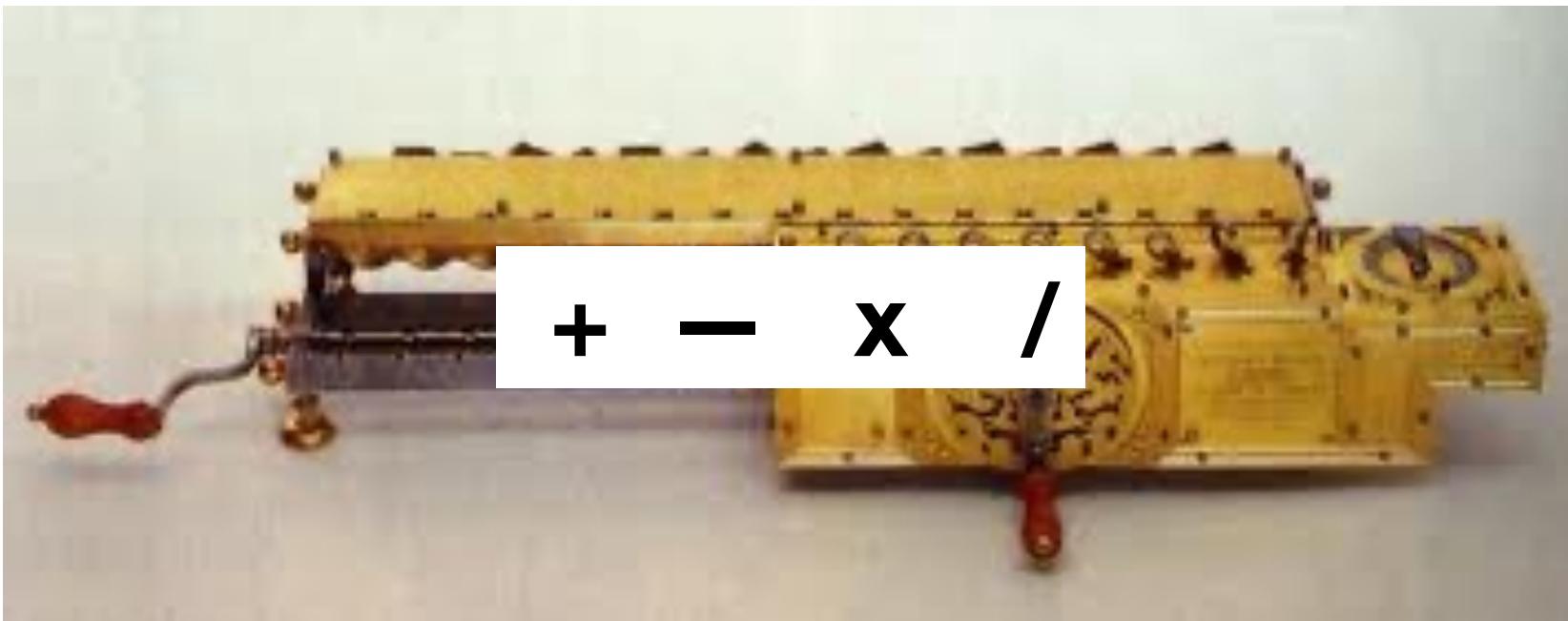
¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?



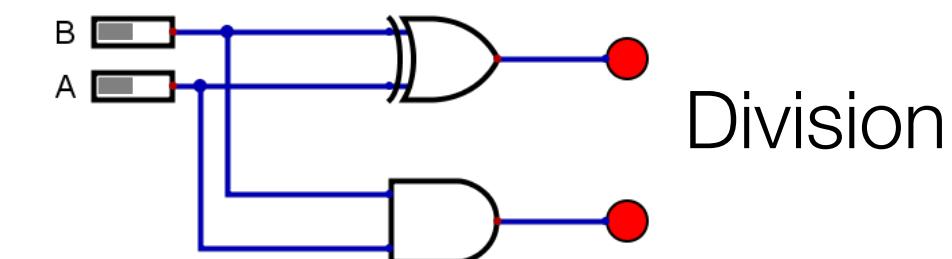
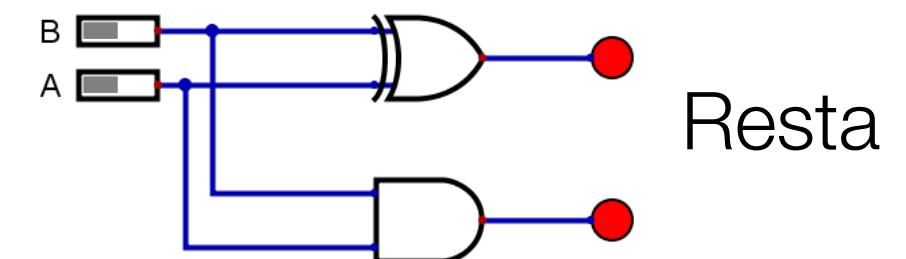
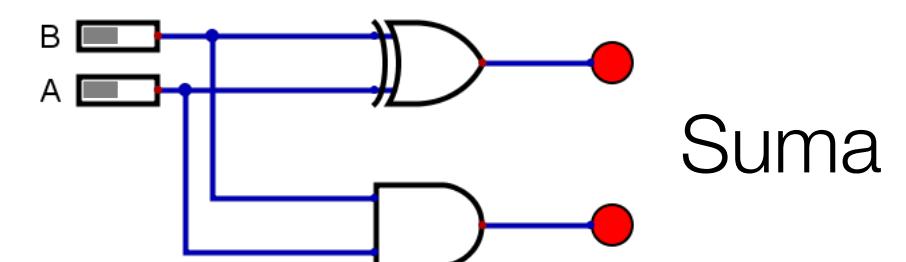


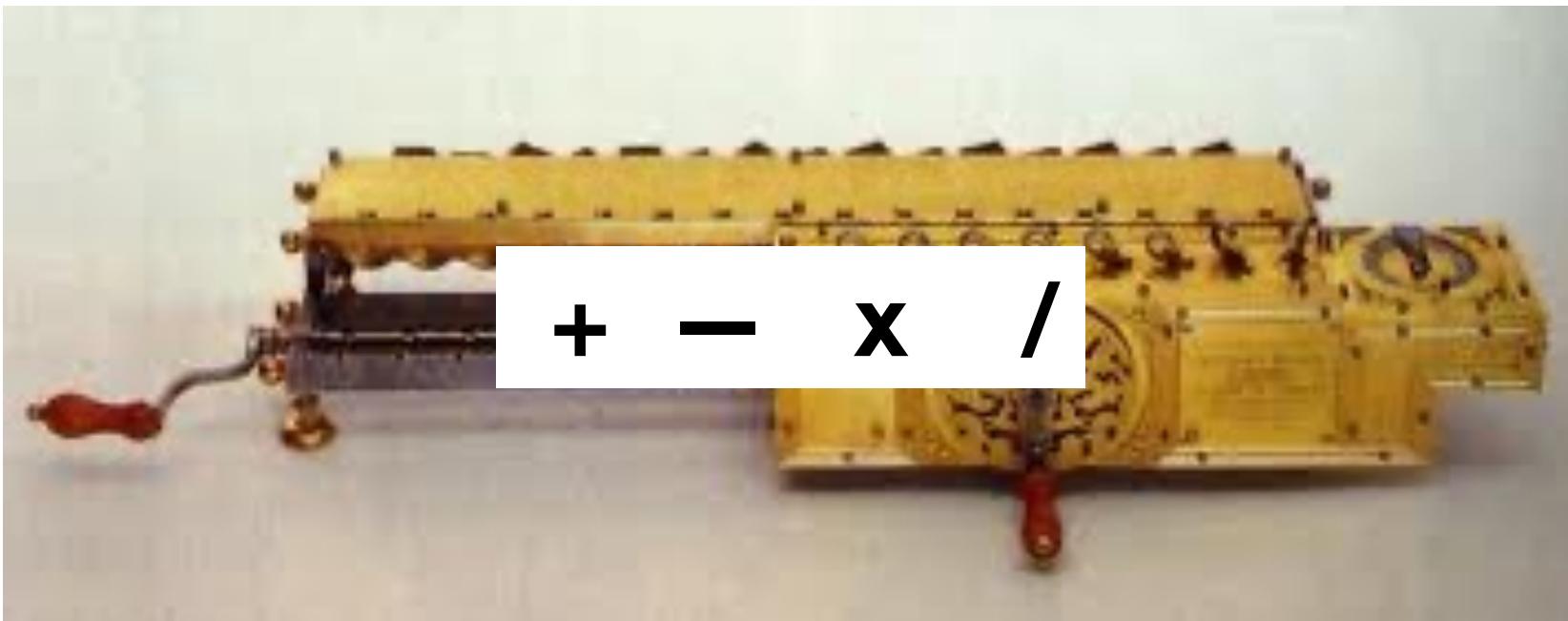
¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?



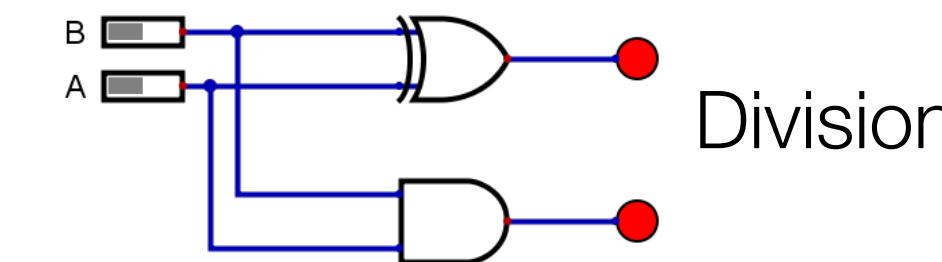
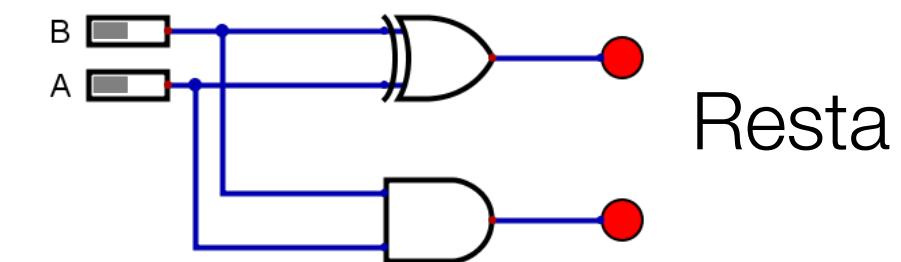
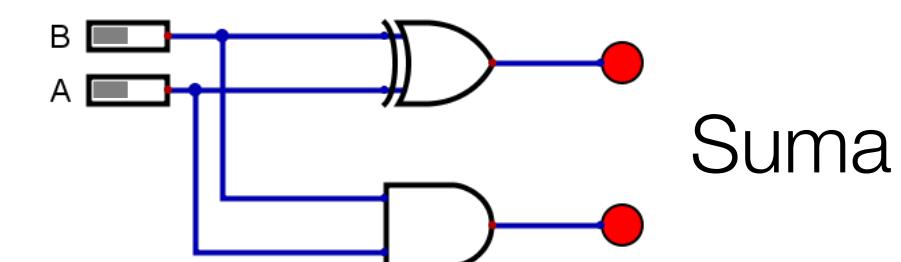


¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?





¿Cómo sería la construcción de una máquina que nos hiciese cálculos sencillos?



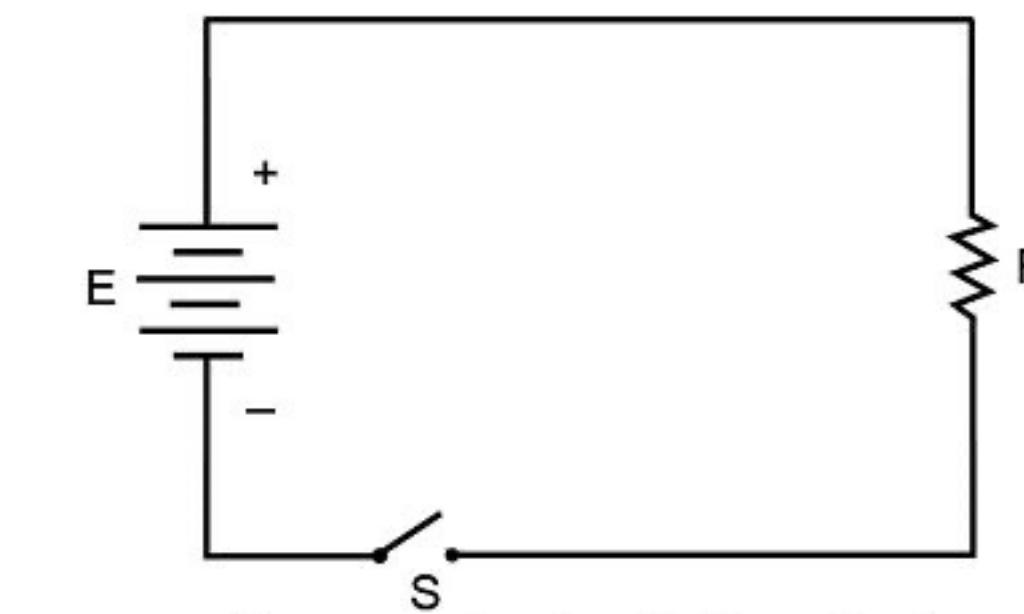
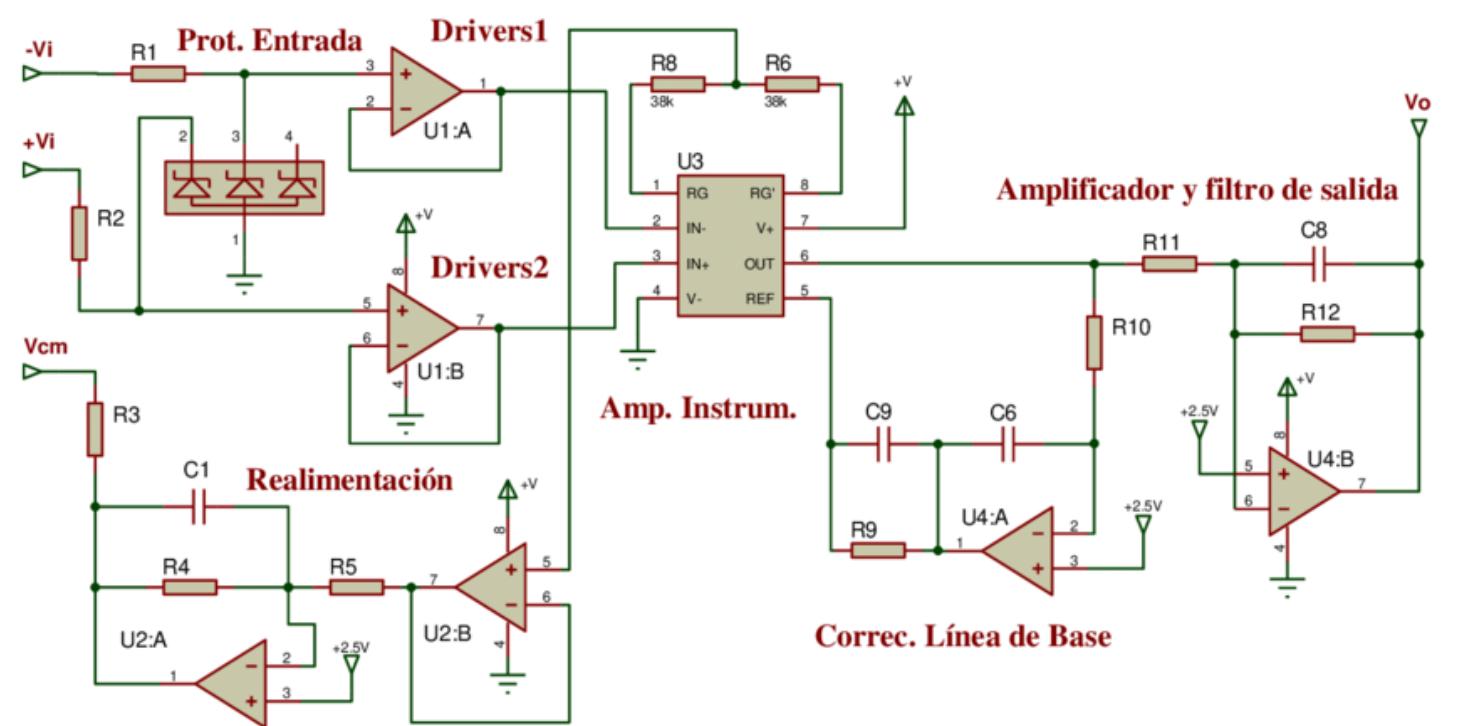
**Ordenador de programa fijo**  
Fixed program computer



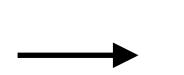
¿Cual sería el circuito ideal?



¿Cuál sería el circuito ideal?



INPUT: diagrama circuito



calculadora

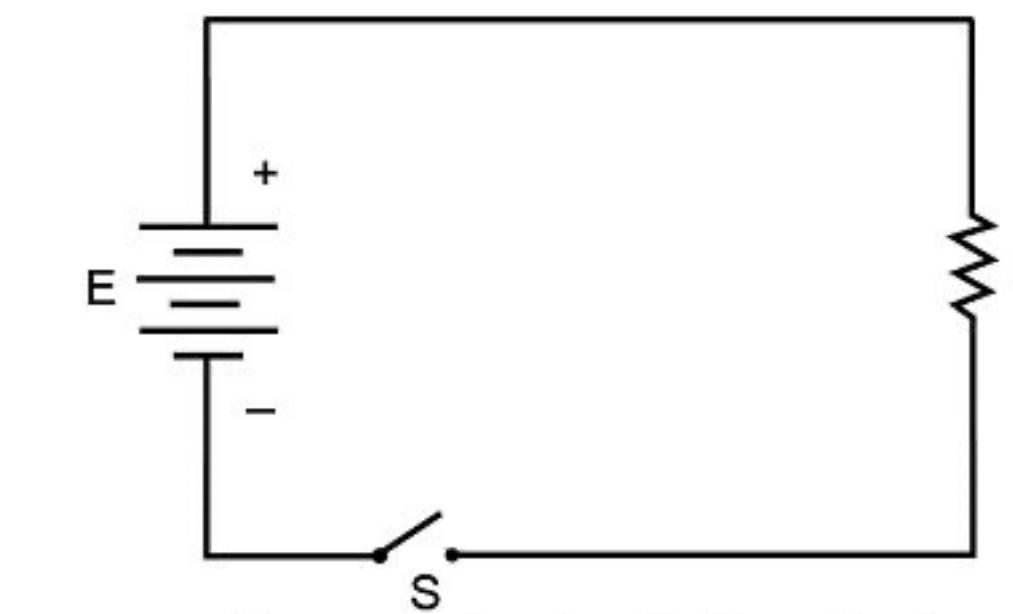
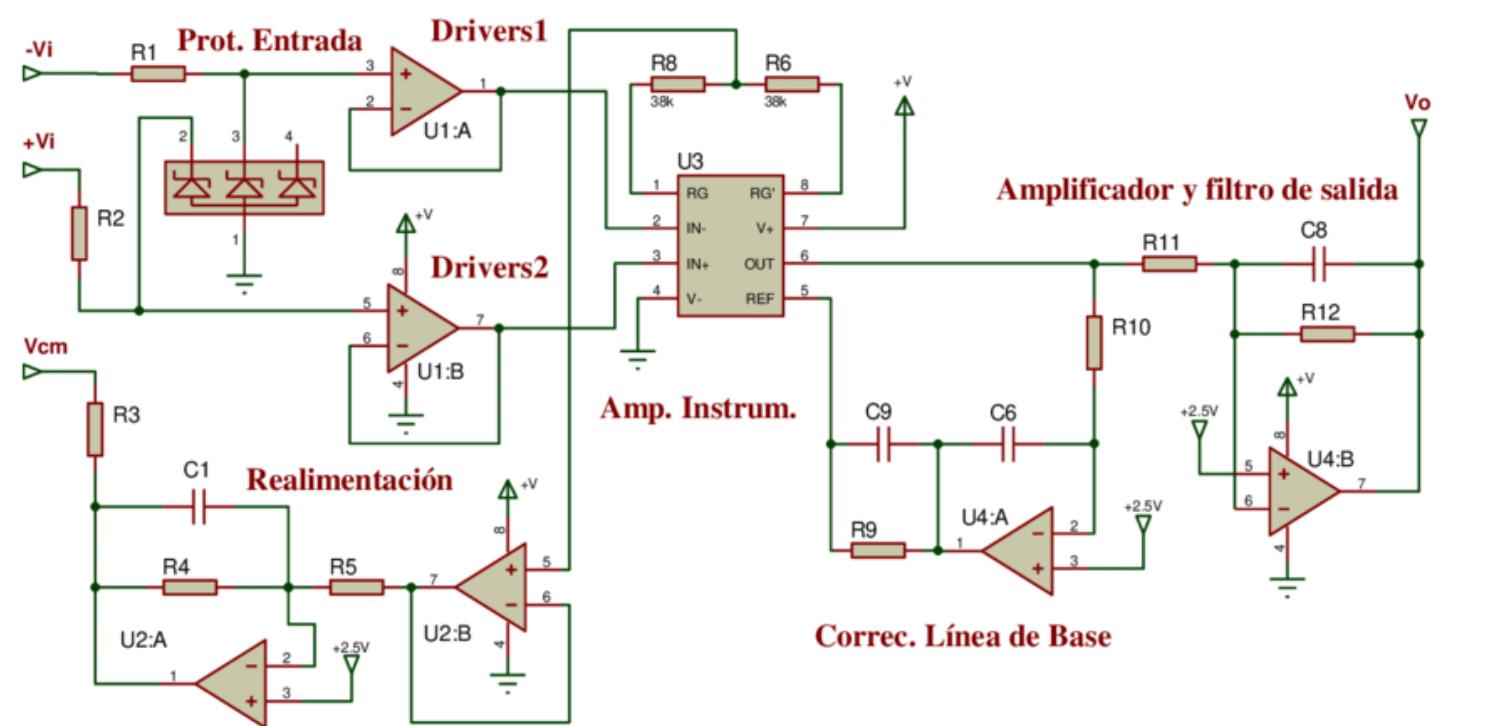
circuito que abre  
una puerta

...

circuito que se reconfigura  
solo para imitar el diagrama



¿Cuál sería el circuito ideal?



INPUT: diagrama circuito



calculadora

circuito que abre  
una puerta

...

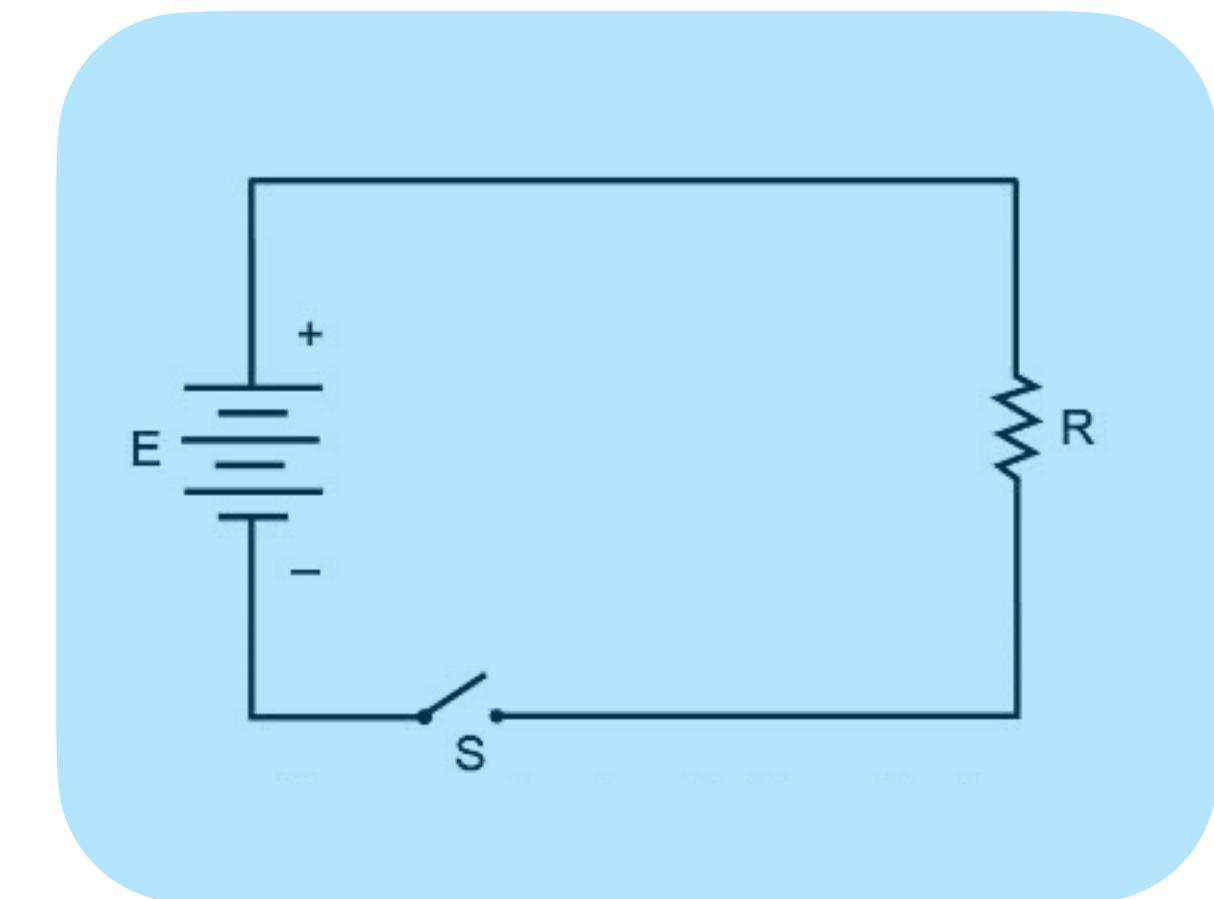
**INTÉPRETE**



¿Cual sería el circuito ideal?

RECETA O SERIE DE INSTRUCCIONES

1. suma A con B
2. escribe el resultado en un documento
3. guarda el documento



Ordenador de programa almacenado

Stored program computer



¿Cual sería el circuito ideal?

RECETA O SERIE DE INSTRUCCIONES

1. suma A con B
2. escribe el resultado en un documento
3. guarda el documento

**MEMORIA**

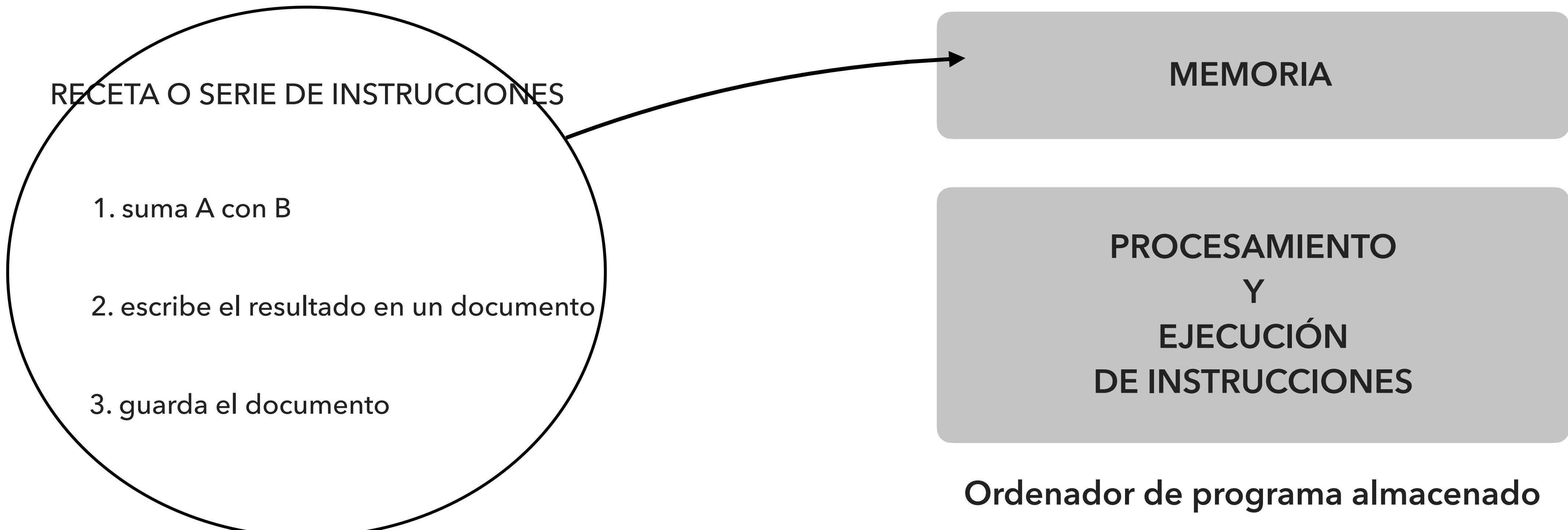
**PROCESAMIENTO  
Y  
EJECUCIÓN  
DE INSTRUCCIONES**

**Ordenador de programa almacenado**

Stored program computer



¿Cuál sería el circuito ideal?





# ¿QUÉ ES UN ORDENADOR?

Datos de entrada  
Input



Datos de salida  
Output



*Hardware*

Parte física del ordenador

*Software*

Parte lógica del ordenador



# ¿QUÉ ES UN ORDENADOR?

Datos de entrada  
Input



Datos de salida  
Output



*Hardware*

Parte física del ordenador

*Software*

Parte lógica del ordenador



# HARDWARE

<b>Dispositivos de entrada</b>	
<b>Dispositivos de salida</b>	
<b>Dispositivos de comunicación</b>	
<b>Dispositivos de almacenamiento</b>	
<b>Dispositivos de cómputo</b>	

MEMORIA

PROCESAMIENTO  
Y  
EJECUCIÓN  
DE INSTRUCCIONES

INPUT

OUTPUT



# HARDWARE

**Primera generación (1946 – 1958):** Máquinas que funcionan con válvulas. Contacto (casi) directo con el hardware de la máquina



# HARDWARE

**Primera generación (1946 – 1958):** Máquinas que funcionan con válvulas. Contacto (casi) directo con el hardware de la máquina

**Segunda generación (1959 – 1964):** Uso de transistores. Permitió la construcción de computadoras más poderosas y menos costosas.



# HARDWARE

- Primera generación (1946 – 1958):** Máquinas que funcionan con válvulas. Contacto (casi) directo con el hardware de la máquina
- Segunda generación (1959 – 1964):** Uso de transistores. Permitió la construcción de computadoras más poderosas y menos costosas.
- Tercera generación (1965 – 1970):** Uso de circuitos integrados.



# HARDWARE

- Primera generación (1946 – 1958):** Máquinas que funcionan con válvulas. Contacto (casi) directo con el hardware de la máquina
- Segunda generación (1959 – 1964):** Uso de transistores. Permitió la construcción de computadoras más poderosas y menos costosas.
- Tercera generación (1965 – 1970):** Uso de circuitos integrados.
- Cuarta generación (1971 – 1988):** Uso de circuitos integrados y transistores a gran escala (más circuitos por unidad de espacio)



# HARDWARE

- Primera generación (1946 – 1958):** Máquinas que funcionan con válvulas. Contacto (casi) directo con el hardware de la máquina
- Segunda generación (1959 – 1964):** Uso de transistores. Permitió la construcción de computadoras más poderosas y menos costosas.
- Tercera generación (1965 – 1970):** Uso de circuitos integrados.
- Cuarta generación (1971 – 1988):** Uso de circuitos integrados y transistores a gran escala (más circuitos por unidad de espacio)
- Quinta generación (1988 – ?):** En vista de la acelerada marcha de la microelectrónica y la nanotecnología, la sociedad industrial se ha dado a la tarea de poner también a esa altura el desarrollo del software y los sistemas con que se manejan las computadoras.



# ¿QUÉ ES UN ORDENADOR?

Datos de entrada  
Input



Datos de salida  
Output



*Hardware*

Parte física del ordenador

*Software*

Parte lógica del ordenador



# SOFTWARE

RECETA O SERIE DE INSTRUCCIONES



PROGRAMA



# SOFTWARE

RECETA O SERIE DE INSTRUCCIONES



PROGRAMA

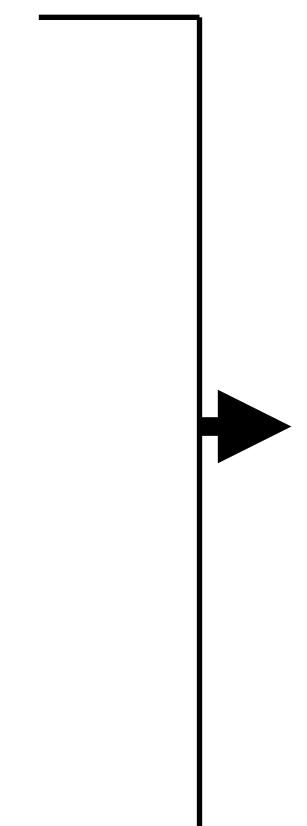
COLECCION DE RECETAS O PROGRAMAS  
INTERRELACIONADOS



PROGRAMA 1

PROGRAMA 2

PROGRAMA 3





# SOFTWARE

RECETA O SERIE DE INSTRUCCIONES



PROGRAMA

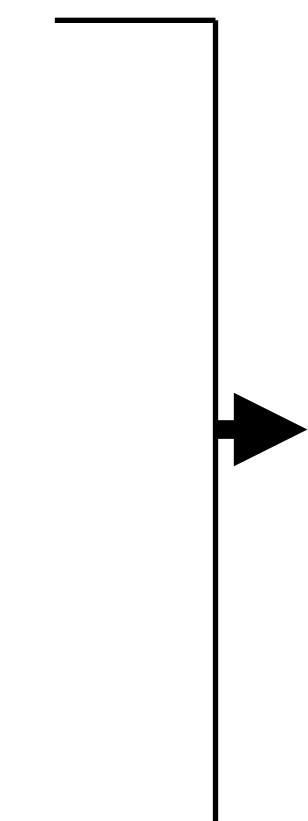
COLECCION DE RECETAS O PROGRAMAS  
INTERRELACIONADOS



PROGRAMA 1

PROGRAMA 2

PROGRAMA 3



**SOFTWARE**



# SOFTWARE: SISTEMA OPERATIVO

**Es el software más fundamental**

Provee una interfaz hardware - usuario

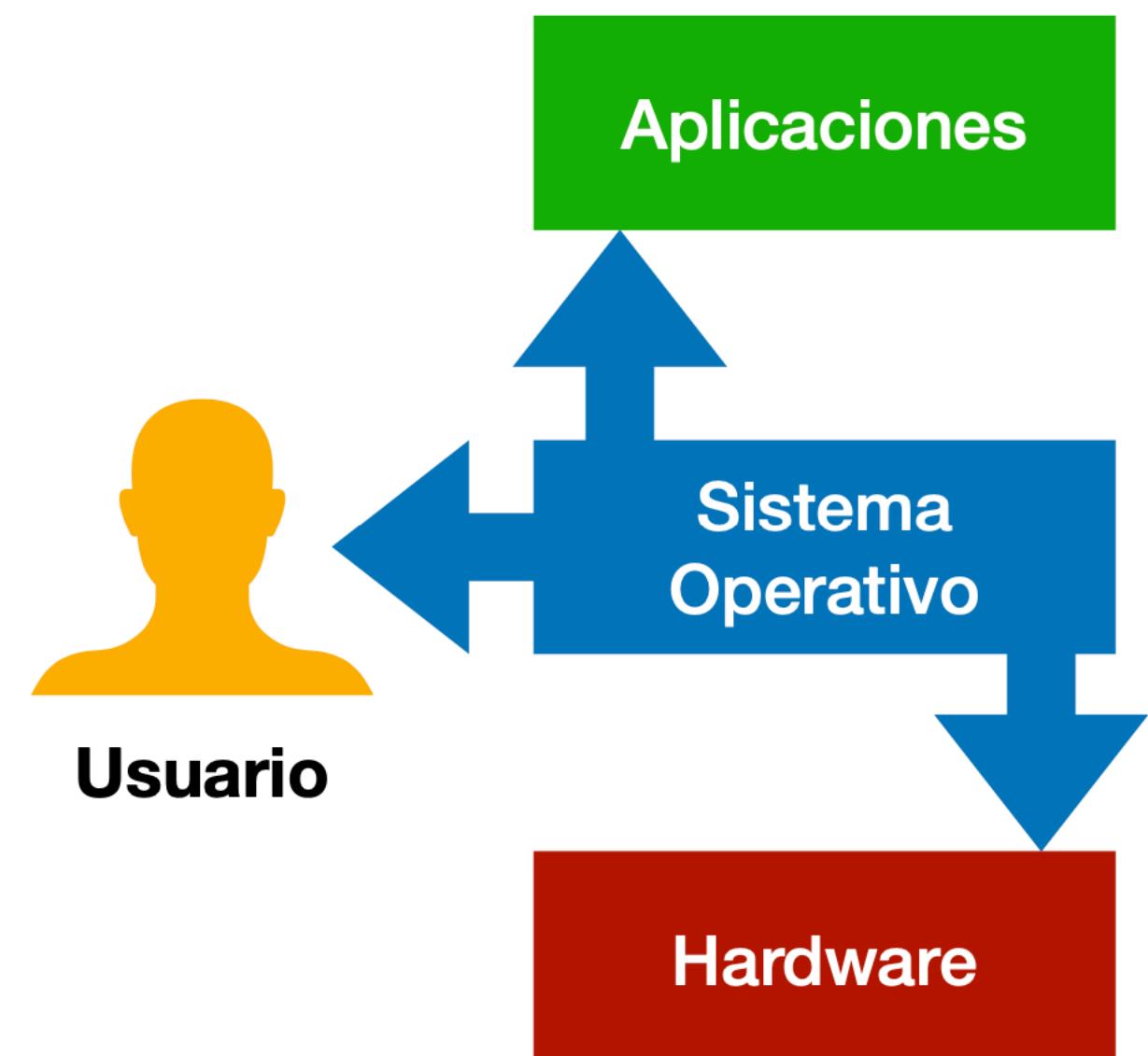
Administra los recursos de la máquina

Organiza los archivos en dispositivos de almacenamiento

Coordina los distintos elementos de hardware

Puede incluir aplicaciones pre-instaladas (internet explorer en Windows)

Ejemplos: MacOS, Windows, Linux

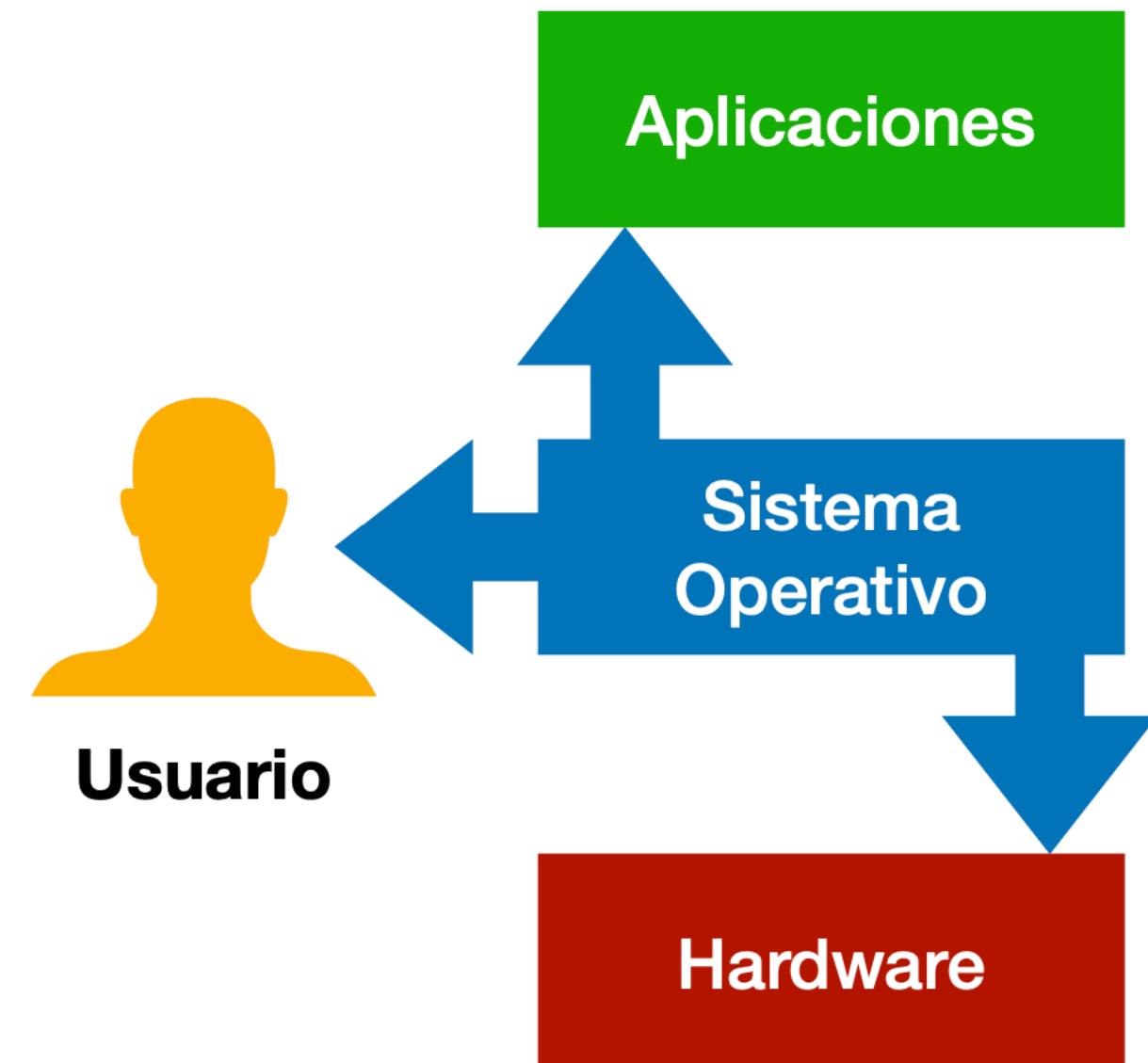




# SOFTWARE: SISTEMA OPERATIVO

## Componentes:

1. Manejo de procesos
2. Manejo de Input y Output
3. Manejo de Memoria
4. Manejo de Sistema de Archivos





# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Idioma artificial diseñado para expresar computaciones



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Idioma artificial diseñado para expresar computaciones

Lenguaje de máquina



0110 1001 1010 1011



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Idioma artificial diseñado para expresar computaciones

Lenguaje de máquina



0110 1001 1010 1011

Lenguaje de bajo nivel



ADD M, N, P



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Idioma artificial diseñado para expresar computaciones

Lenguaje de máquina



0110 1001 1010 1011

Lenguaje de bajo nivel



ADD M, N, P

Lenguaje de alto nivel



P = M + N



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Lenguaje de máquina



0110 1001 1010 1011

CÓDIGO BINARIO

Pasa corriente [1]  
o  
no pasa corriente [0]

## VENTAJAS:

Velocidad de ejecución insuperable

## DESVENTAJAS:

Difícil y lento de codificar  
Difíciles de comprender y mantener  
Dependen del ordenador en el que se ejecuten  
Son poco fiables

# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Lenguaje de bajo nivel



ADD M, N, P

ENSAMBLADOR



Fase de traducción

0110 1001 1010 1011

PROGRAMA OBJETO

## VENTAJAS:

Más fáciles de codificar  
que el lenguaje máquina

Relativamente rápidos en  
la ejecución

## DESVENTAJAS:

Difícil y lento de codificar  
Dependen del ordenador en el  
que se ejecuten



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Lenguaje de alto nivel



$$P = M + N$$

PYTHON  
JAVA  
C++  
SOLIDITY  
...

## VENTAJAS:

- Más fácil escribir y entender los programas
- Fácil hacer modificaciones y mantener el software al día
- No dependen del ordenador donde se ejecuten
- Tiempo de formación necesario es menor

## DESVENTAJAS:

- Ocupan más espacio en memoria
- El tiempo de ejecución es mayor



# SOFTWARE: LENGUAJES DE PROGRAMACIÓN

## Lenguaje de bajo nivel

Ensamblador:

```
MOV1 R1 = Type  
LD4   R2 = [R1]  
;;  
cmp.eq P1, P2 = 'a', R2  
cmp.eq P3, P4 = 'b', R2  
;;  
(P1) Add R2 = 10, R2  
(P3) Add R2 = 20, R2  
;;  
st4 (R1) = (R2)  
default::
```

## Lenguaje de alto nivel

C:

```
SWITCH TYPE)  
{ case 'a':  
    type = type + 10  
    break;  
  case 'b':  
    type = type + 20;  
    break;  
  default:  
    break;}
```



# SOFTWARE: CREACIÓN DE UN PROGRAMA



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

1. Nos acercamos a la casa



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

1. Nos acercamos a la casa
2. Abrimos la puerta



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

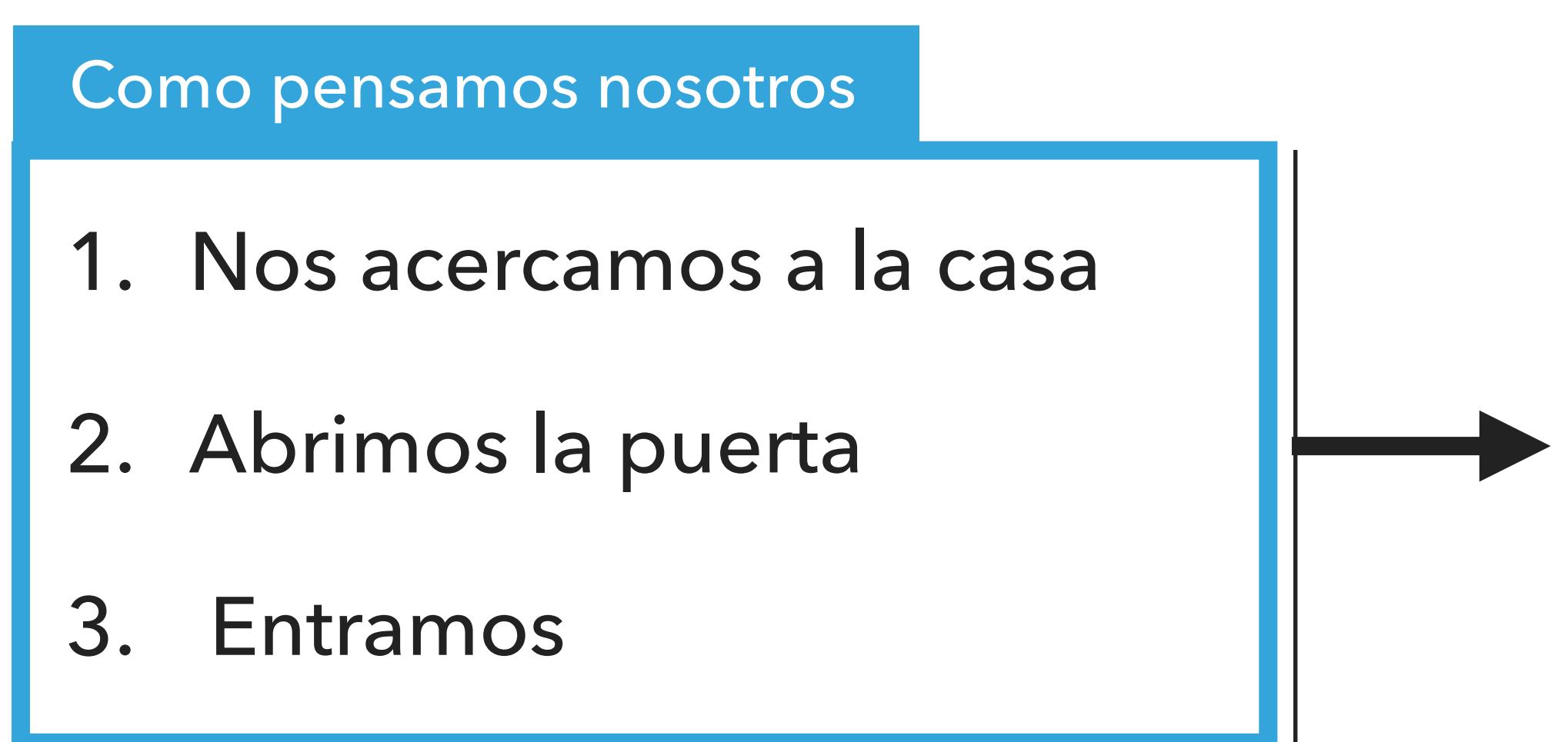
Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

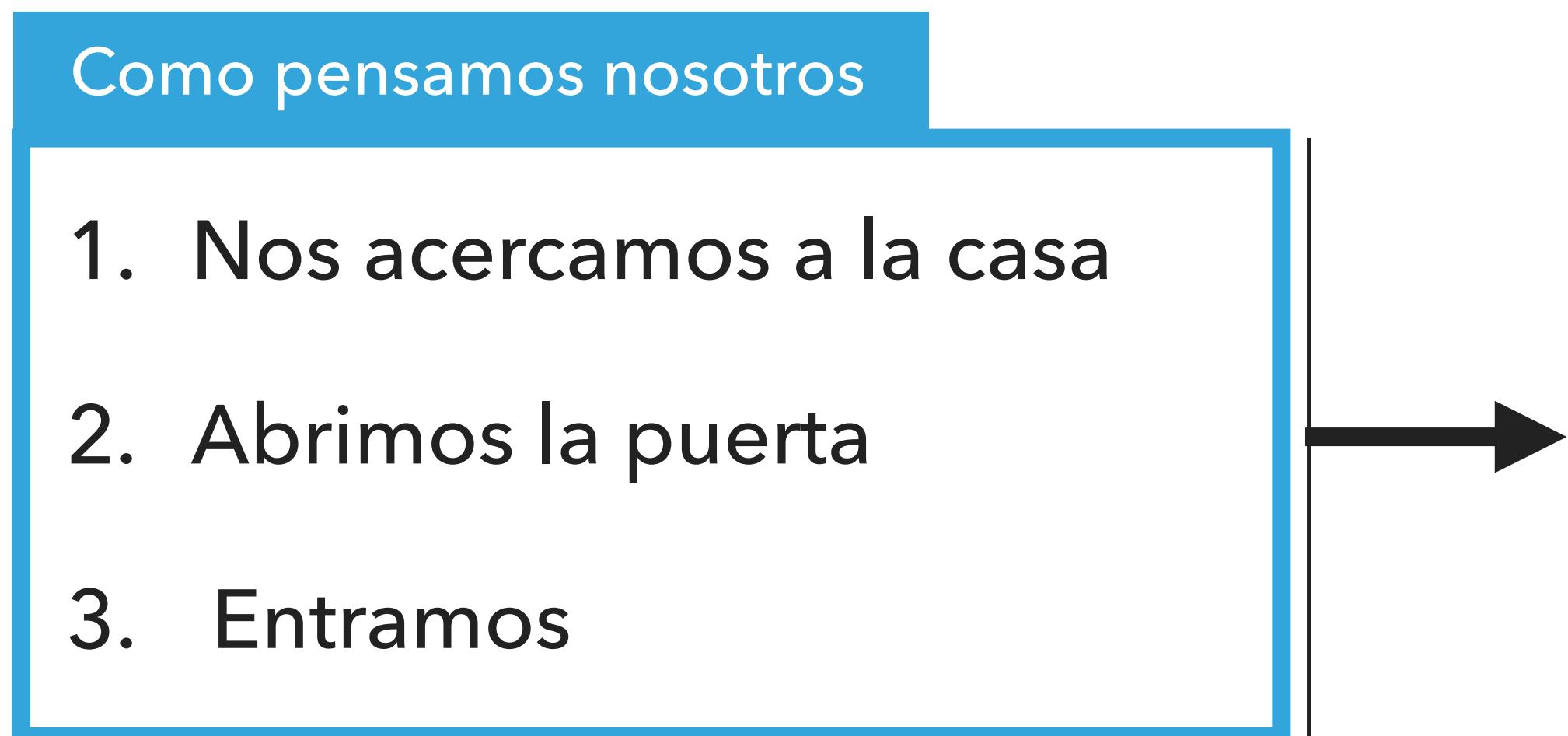


1. Nos acercamos a la puerta de la casa



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA





# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano





# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante

# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados
12. Apoyamos la mano en la puerta



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados
12. Apoyamos la mano en la puerta
13. Empujamos la puerta hasta que suene 'clack'



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

## Como pensamos nosotros

1. Nos acercamos a la casa
2. Abrimos la puerta
3. Entramos



## Lógica de la máquina

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados
12. Apoyamos la mano en la puerta
13. Empujamos la puerta hasta que suene 'clack'



# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Sentencia Declarativa

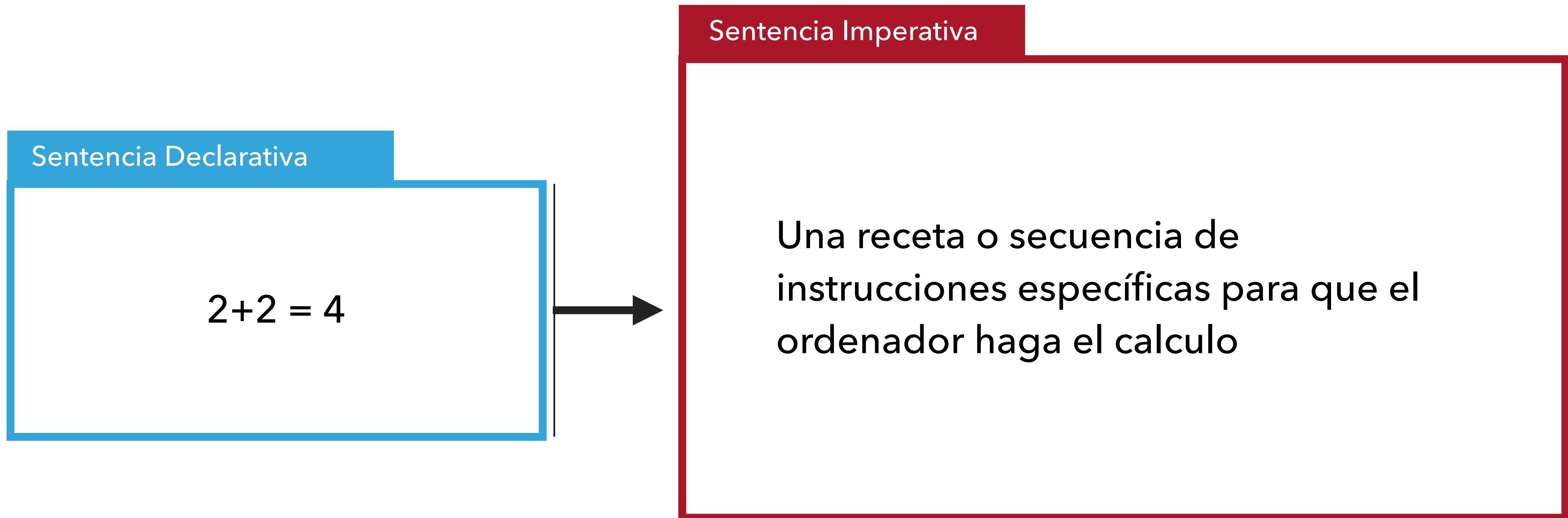
Entremos en la casa

## Sentencia Imperativa

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo el juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga click
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados
12. Apoyamos la mano en la puerta
13. Empujamos la puerta hasta que suene 'clack'

# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA





# SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

## Sentencia Declarativa

Entremos en la casa



## Sentencia Imperativa

1. Nos acercamos a la puerta de la casa
2. Buscamos en el bolsillo del juego de llaves
3. Sacamos la mano con la llave de la casa
4. Levantamos la mano
5. Metemos la llave en la cerradura
6. Giramos la mano hasta que haga **click**
7. Apoyamos la mano en la puerta
8. Empujamos hacia dentro
9. Sacamos la llave de la cerradura
10. Damos unos pasos hacia delante
11. Giramos en 180 grados
12. Apoyamos la mano en la puerta
13. Empujamos la puerta hasta que suene '**clack**'

CONQUER  
**BLOCKS**