

ORGANIZACIÓN DE COMPUTADORES

# Sistema Binario y Compuertas Lógicas

Luis Astudillo y Manolo Iñiguez

# Contexto y Metodología

- Por qué realizamos este tema?
- Cuál es el objetivo?
- Cómo lo desarrollamos?
- Experimentos que hicimos

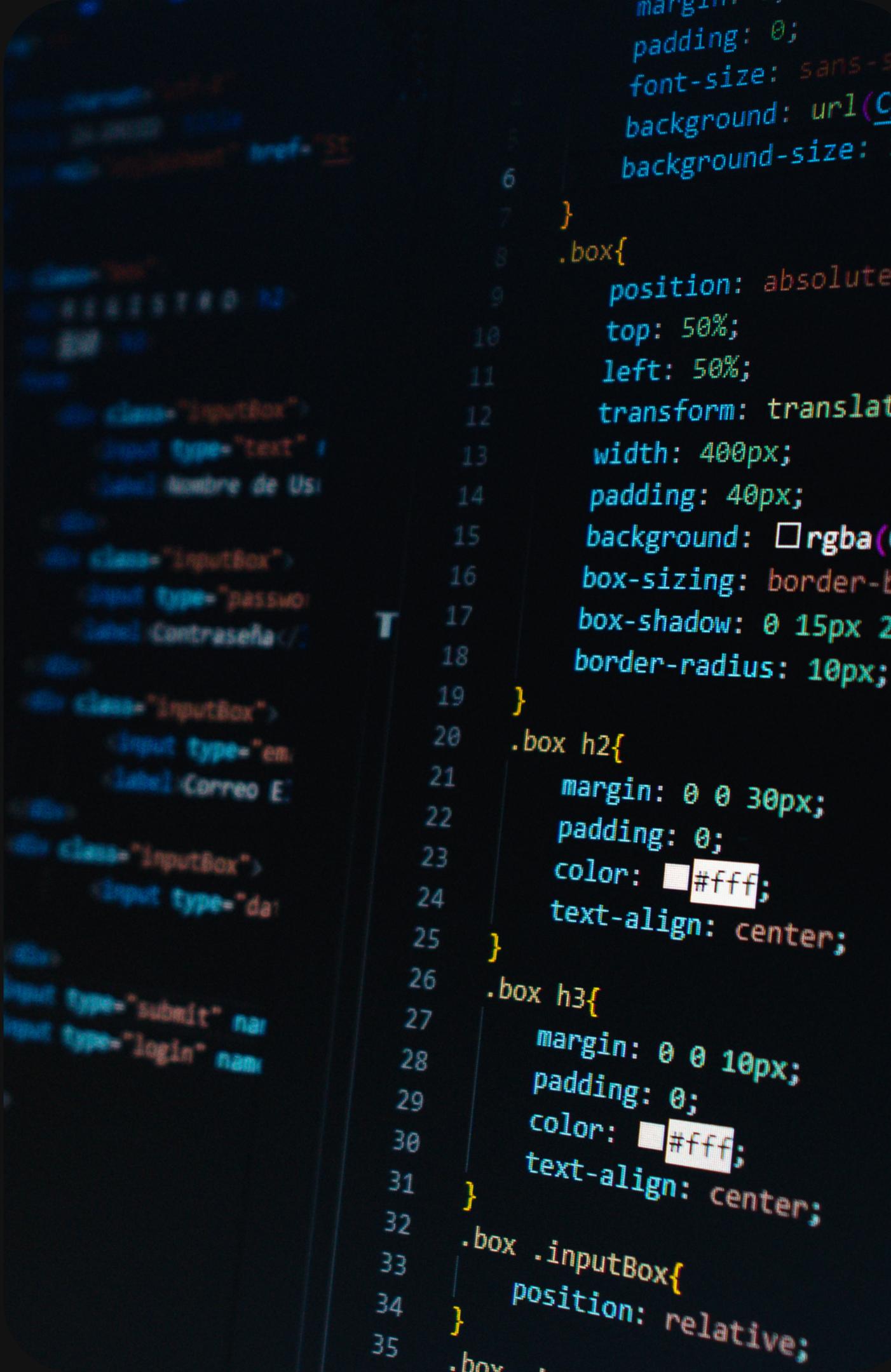


# Sistema binario

# Sistema numérico de dos dígitos, 0 y 1.

# Cálculos matemáticos y transmitir información.

Es el método más exitoso y que actualmente se utiliza.



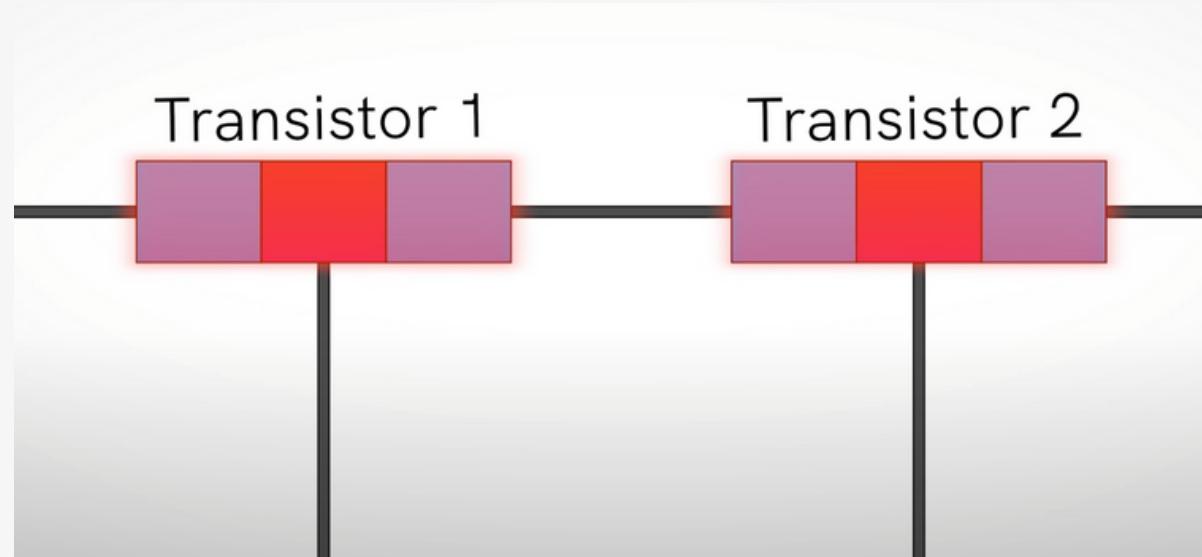
**Como operar  
aritméticamente  
con estas señales  
eléctricas?**

# Compuertas lógicas

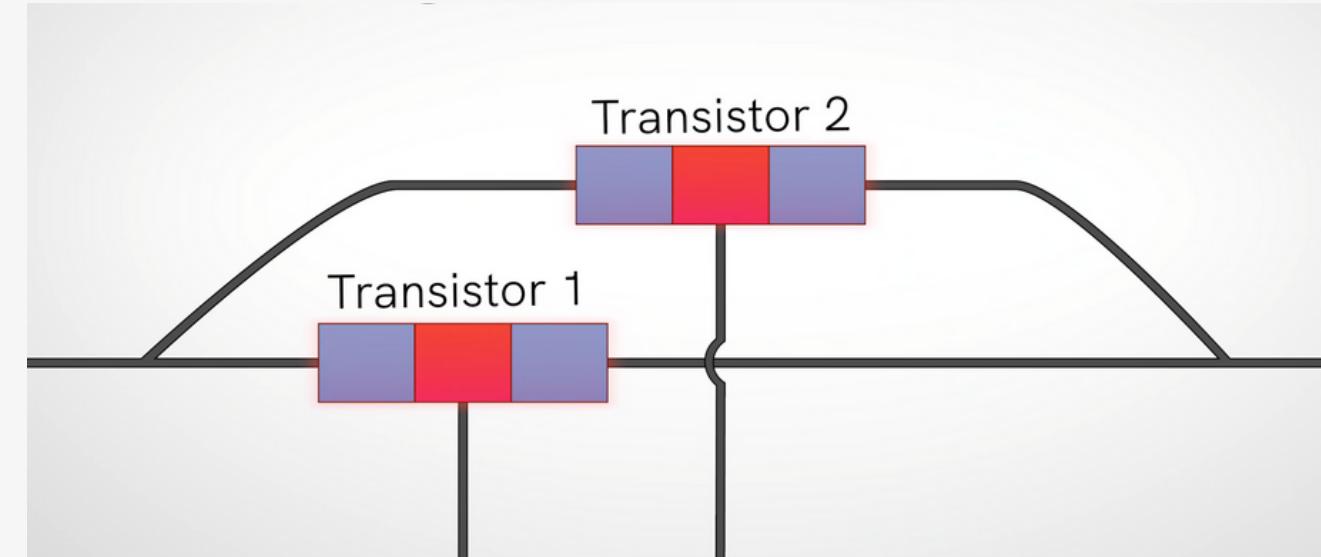
- AND, NAND, NOT, OR, NOR, XOR, XNOR
- Funcionan por medio de transistores que actúan como puentes.



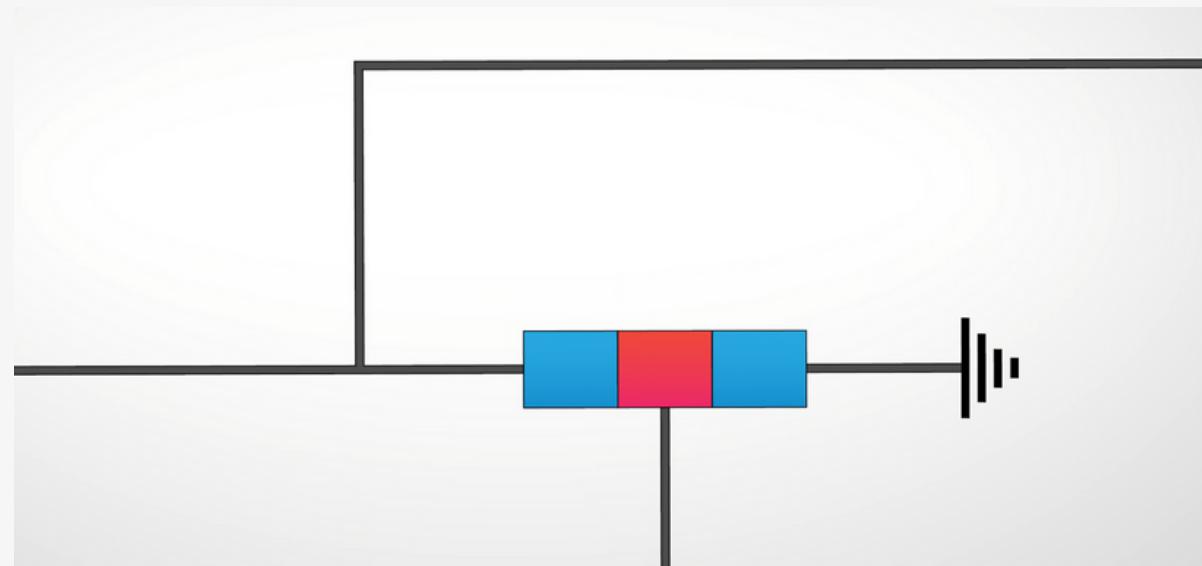
# AND



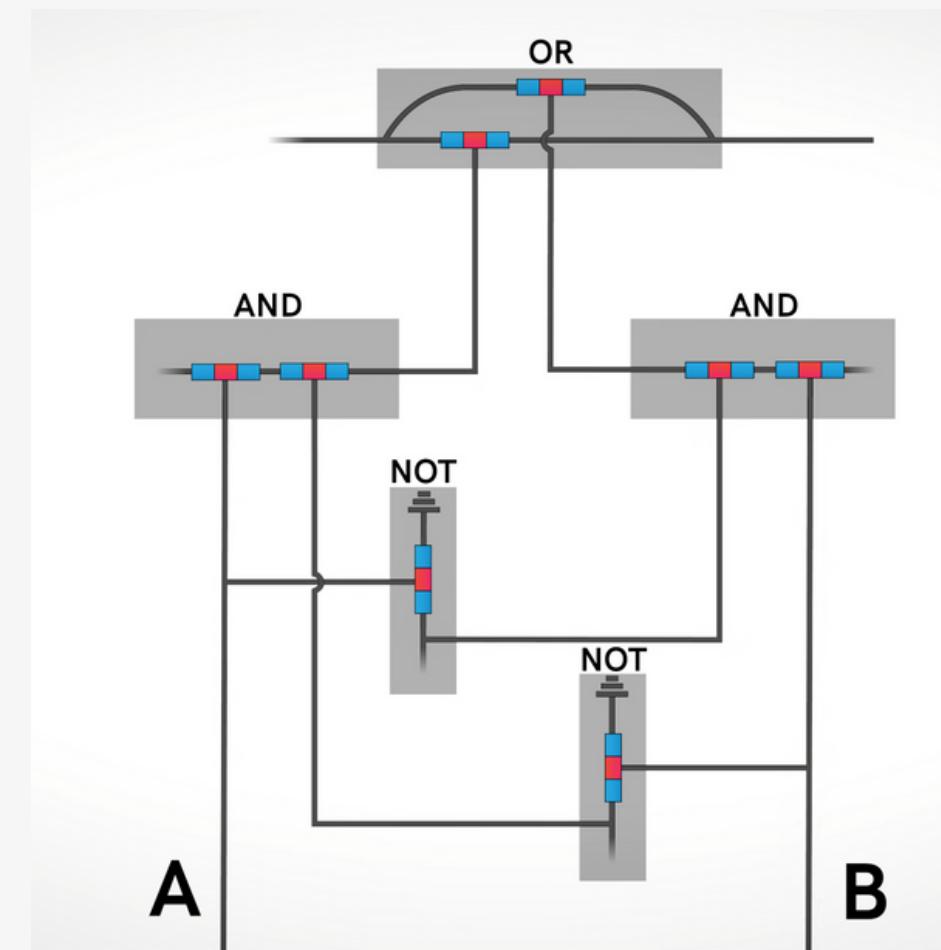
# OR



# NOT



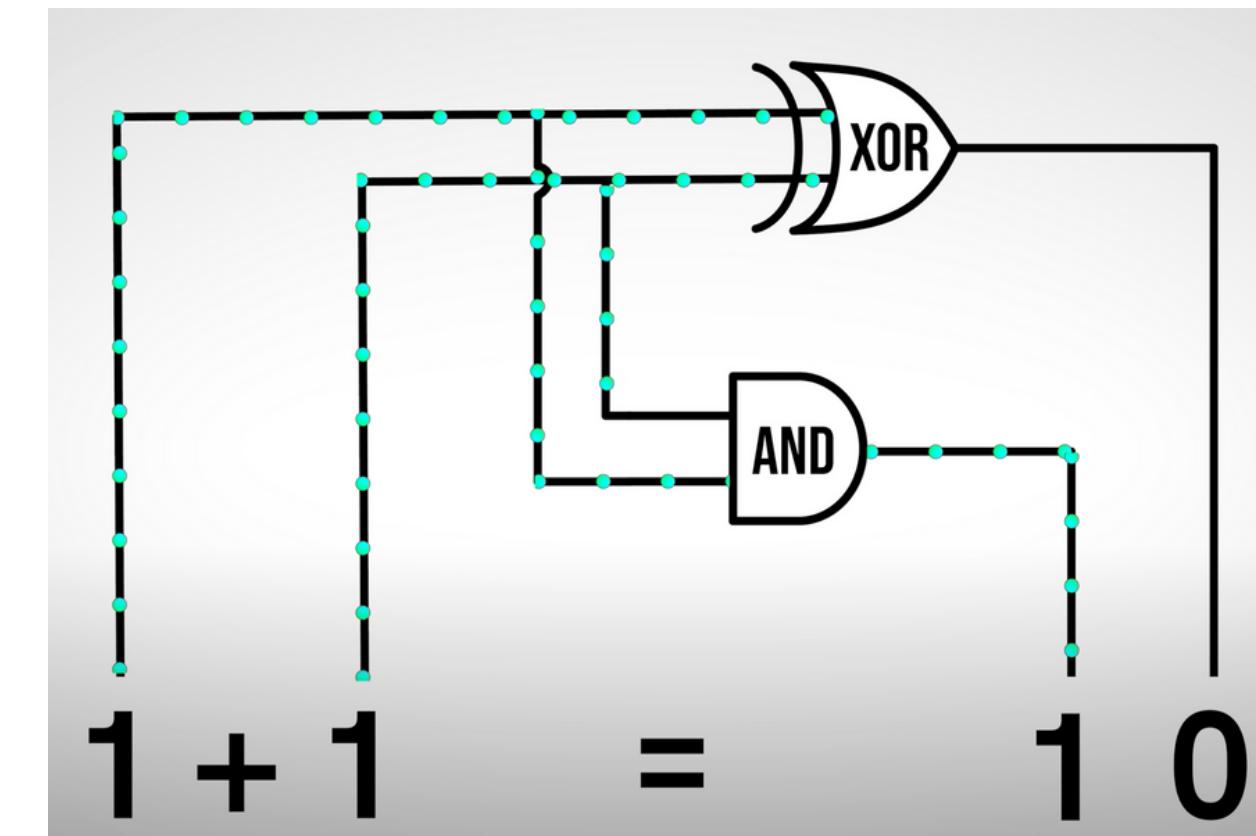
# XOR



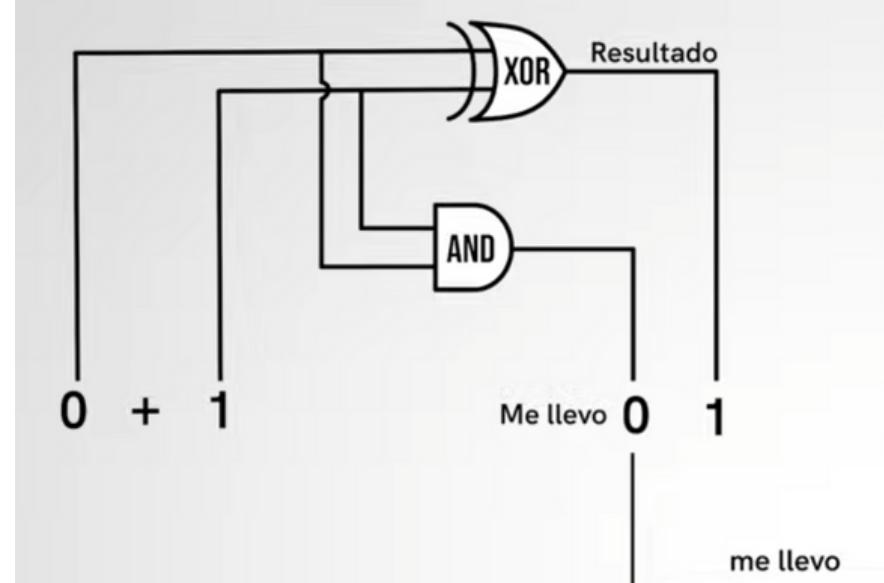
# Maquina para Sumar

La forma más sencilla es utilizar la compuerta *XOR*

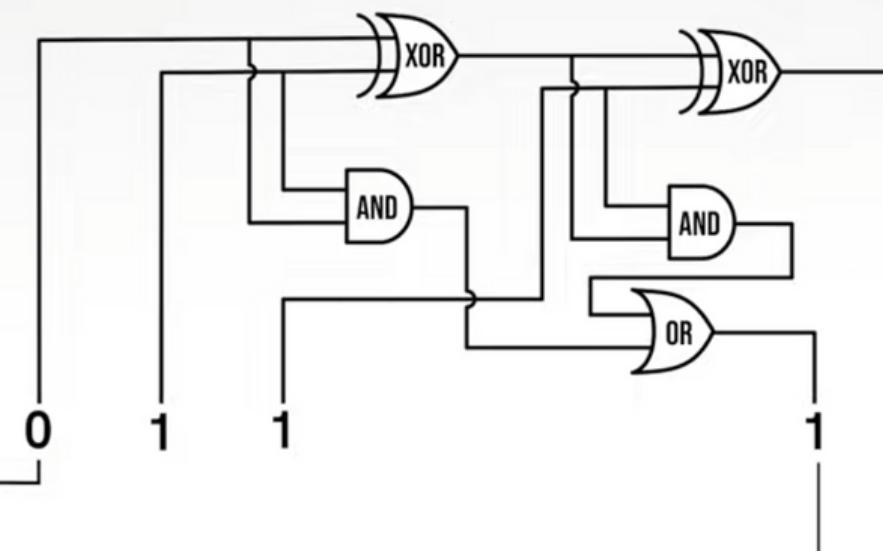
Sumador medio



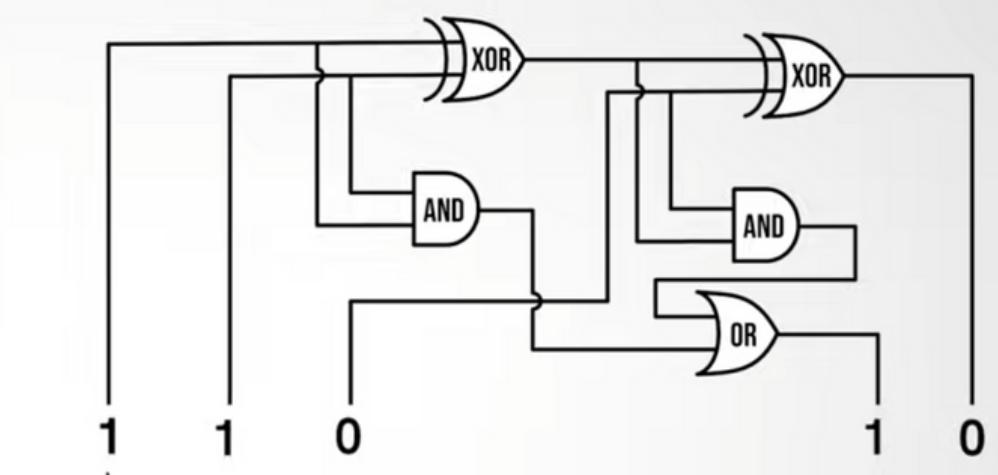
- $0 + 0 = 0$



- $1 + 0 = 1$

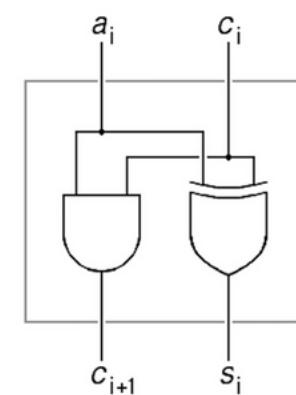


- $0 + 1 = 1$



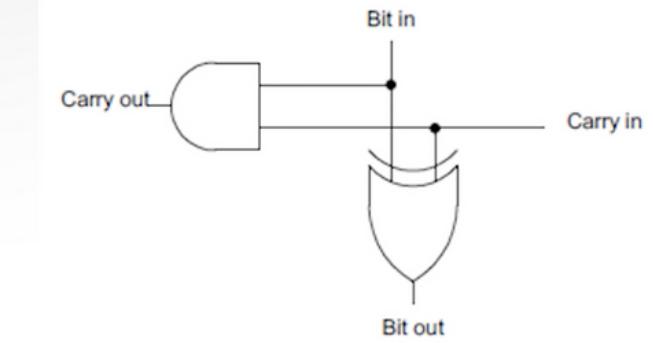
- $1 + 1 = 10$

# Otras operaciones



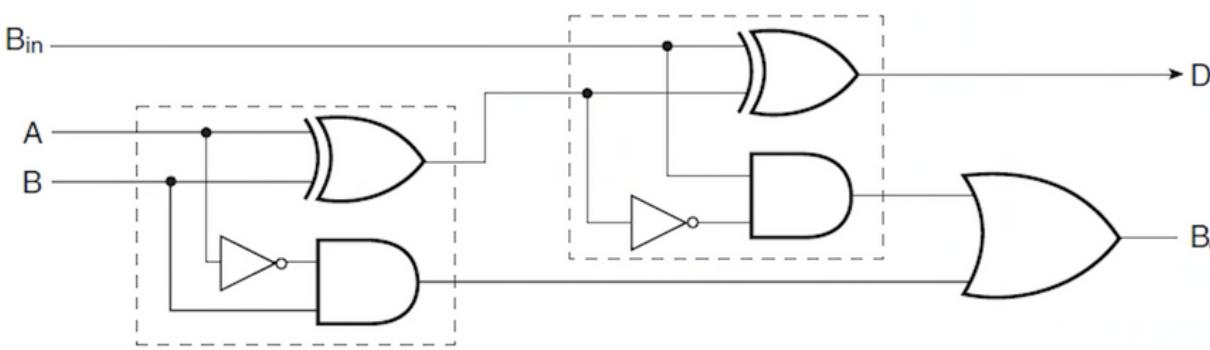
$$Y = A + 1$$

Incremento (INC)



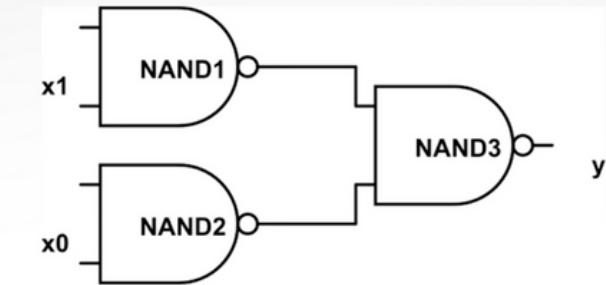
$$Y = A + \bar{1} + 1$$

Decremento (DEC)



$$Y = A - B$$

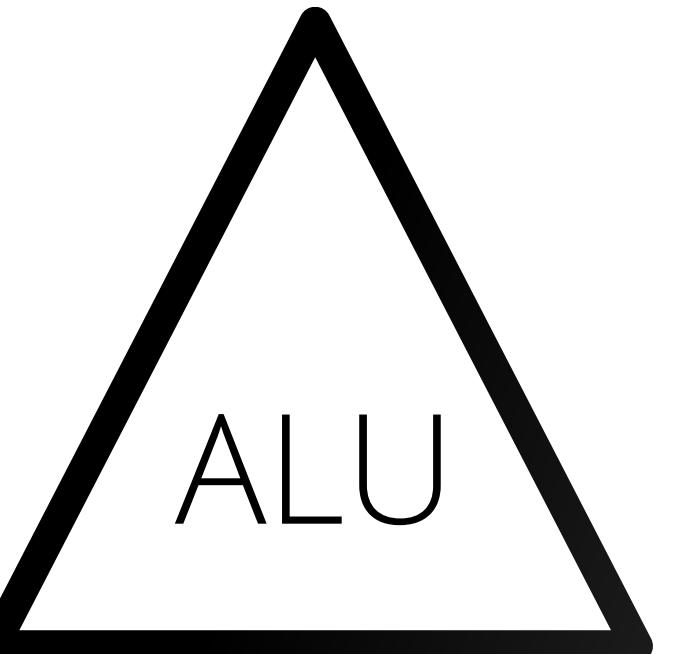
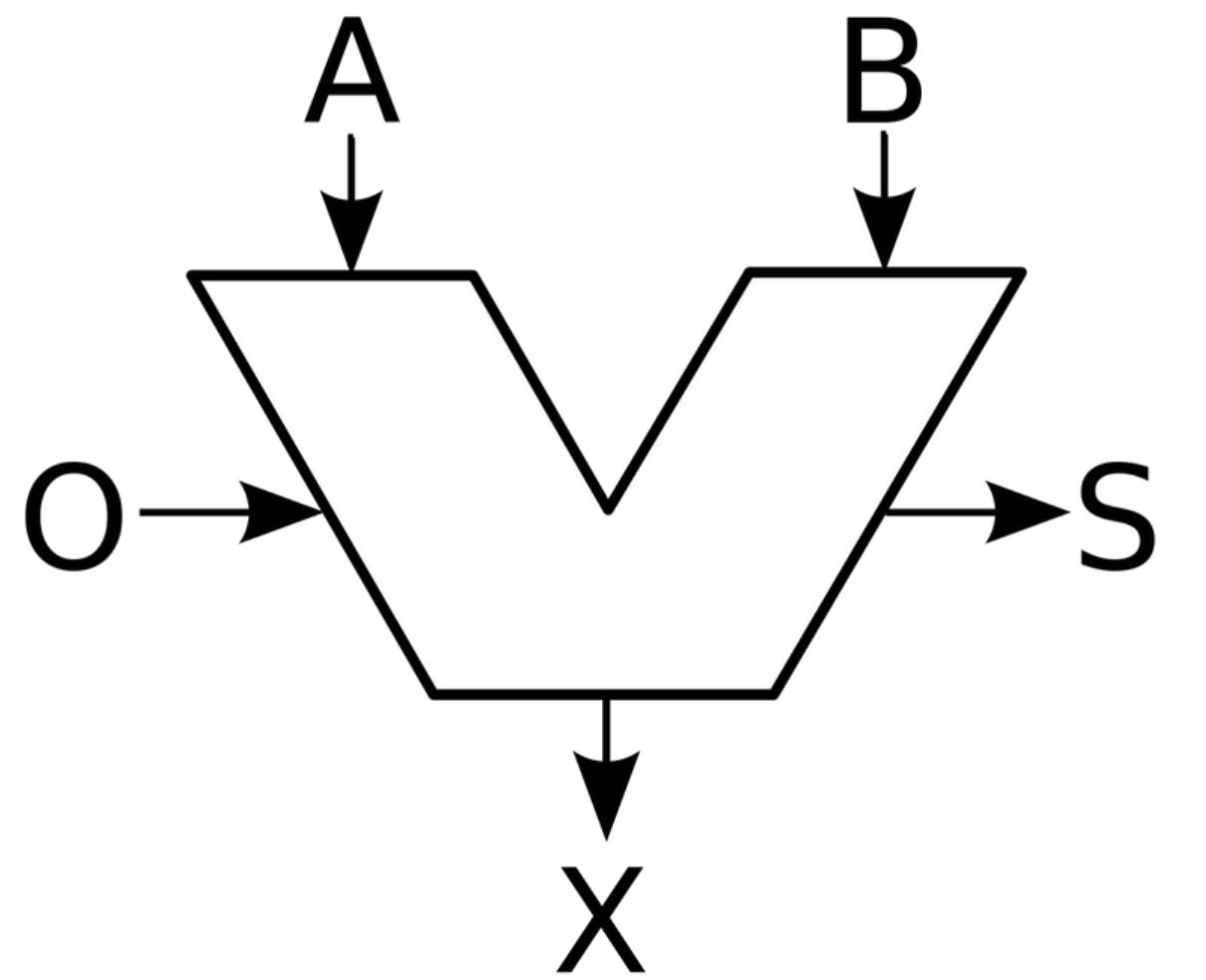
Resta v(SUB)



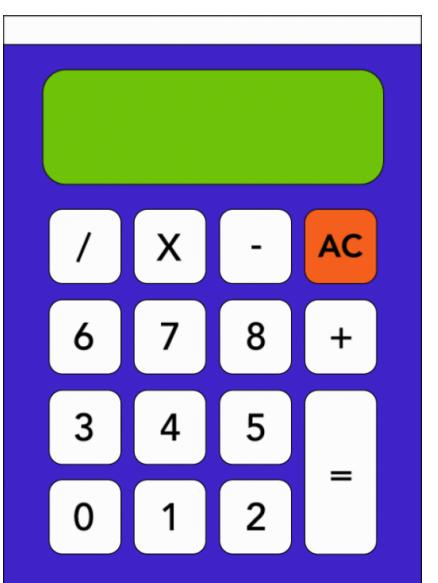
$$Y = 0 + \bar{A} + 1$$

Negación (NEG)

# ALU (Arithmetic logic unit)



MÁQUINAS DE  
OPERACIONES



COMPUERTAS  
LÓGICAS





# PROCESAR TEXTO

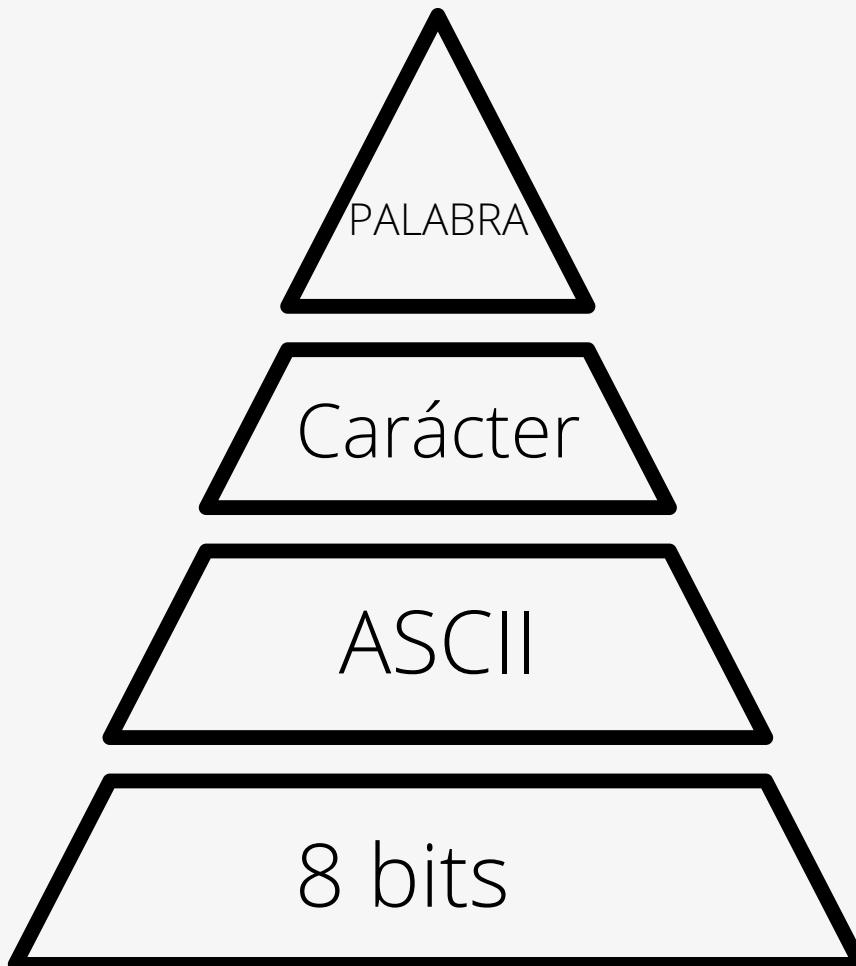
En ascii se utilizan 8 bits por cada letra por lo que tendríamos un número binario entre 0 y 255.

## Tabla ASCII (Solo Letras Minúsculas)

a - 97	h - 104	o - 111	v - 118
b - 98	i - 105	p - 112	w - 119
c - 99	j - 106	q - 113	x - 120
d - 100	k - 107	r - 114	y - 121
e - 101	l - 108	s - 115	z - 122
f - 102	m - 109	t - 116	
g - 103	n - 110	u - 117	



# Ejemplo realizado en java



```
11 import java.io.IOException;
12 import java.io.InputStreamReader;
13
14
15 public class Main {
16
17     public static ArrayList<String> binaryArray = new ArrayList<String>();
18
19     public static void main(String[] args) throws IOException {
20
21         BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
22         System.out.print("Ingrese un texto\n");
23
24
25         String auxTexto = br.readLine();
26
27         for(int i = 0; i < auxTexto.length() ; i++){
28
29             char character = auxTexto.charAt(i);
30             int ascii = (int) character;
31             String binary = Integer.toBinaryString(ascii);
32
33             binaryArray.add(binary);
34
35             System.out.println("Letra: " + character + " ASCII: " + ascii + " Binario: " + binary + "\n");
36         }
37
38         System.out.print(auxTexto + " en binario : ");
39         for(int i = 0; i < binaryArray.size(); i++){
40             System.out.print(binaryArray.get(i) + " ");
41         }
42     }
43 }
```

Main.java

Compile Messages jGRASP Messages Run I/O Interactions

Stop Clear Copy

# Cómo entiende las imágenes el computador?





# Pero qué es el RGB?

## COMPOSICIÓN DE COLORES

Red o rojo

Green o verde

Blue o azul



Red: 255  
Green: 0  
Blue: 0

Binary:  
`111111100000000000000000`



Red: 0  
Green: 255  
Blue: 0

Binary:  
`000000001111111000000000`



Red: 0  
Green: 0  
Blue: 255

Binary:  
`000000000000000011111111`



Librerías: BufferedImage y Color.

# Ejemplo realizado en java



# Conclusiones



Los bits (dígitos binarios) son la base de todo



Capas según su nivel de abstracción



Un antes y un después

# Preguntas?

