

# Fundamentos en Computación

¡La mejor herramienta de los científicos!

---

Luis Fernando Quiroga Peláez  
Correo: [luis.quiroga@udea.edu.co](mailto:luis.quiroga@udea.edu.co)

# Información General del Curso



- **Clases teóricas**

**G11 y G12:** L 6-8 (aula 3-107)

**G5 y G6:** L 8-10 (aula 5-307)

- **Clases prácticas**

**G5:** W 16-18 (aula 6-125), Julián Calle

**G6:** V 16-18 (aula 6-125), Julián Calle

**G11:** W 6-8 (aula 6-125), Mariano Celada

**G12:** V 6-8 (aula 6-125), Daniel Ocampo

# Información General del Curso

## Cómo será la evaluación del curso?

Evaluación		
Teoría 50%	Unidad 1	Examen 1: 10%
	Unidad 2	Examen 2: 20%
	Unidad 3	Examen 3: 20%
Taller 50%	Unidad 1	Taller 1: 10% (Trabajo en el taller y en la casa)
	Unidad 2	Taller 2: 10% (Trabajo en el taller y en la casa)
	Unidad 3	Taller 3: 10% (Trabajo en el taller y en la casa)
	Sustentación oral: 20 %	Unidad 2 y 3

# Información General del Curso

## Cómo nos vamos a comunicar?



El curso usará las diferentes herramientas de google para la comunicación:

1. Google Classroom (Código 5ackzvj)
2. Correo electrónico

[luis.quiroga@udea.edu.co](mailto:luis.quiroga@udea.edu.co)

# Información General del Curso

## Cuáles son los temas principales del curso?

El curso se dividirá en los siguientes 3 temas principales:

1. El computador y Pseudo-código
2. Python
3. Introducción a la modelación



# Introducción

“Vivimos en una sociedad exquisitamente dependiente de la ciencia y la tecnología, pero en la que nadie sabe nada acerca ni de la ciencia ni de la tecnología. Ello constituye una fórmula para el desastre”

---

Carl Sagan

La computación  
es una  
herramienta  
fundamental  
para el científico



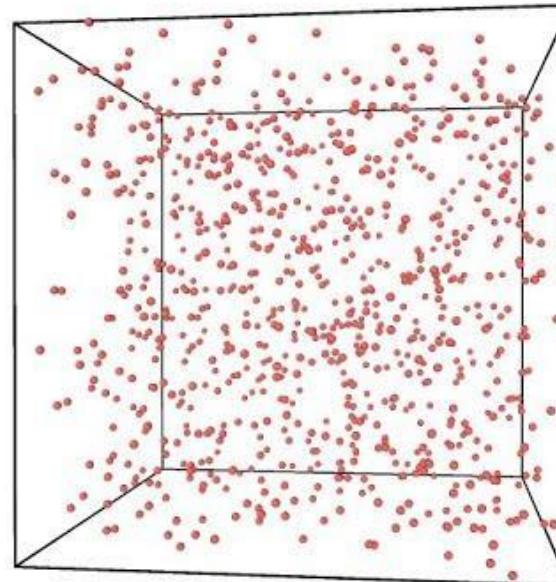
# La computación en Física y Astronomía

¿Por qué un físico o un astrónomo necesita un computador?

- Muchos cálculos

Ejemplo: Calcular la fuerza gravitacional entre las partículas de 1 mol de gas

$$\vec{F} = \frac{Gm_1m_2}{r^2}\hat{r}$$



Número de operaciones:

$$\frac{10^{23} \times (10^{23} - 1)}{2}$$

# Algunos ejemplos

- [https://www.youtube.com/watch?v=\\_3uQqrrBcrQ](https://www.youtube.com/watch?v=_3uQqrrBcrQ)
- <https://www.youtube.com/watch?v=O8tULZHlvtM>
- <https://www.youtube.com/watch?v=LLRfVfjt-ls>
- <https://www.youtube.com/watch?v=tl4mx0TtaAc>
- <https://www.youtube.com/watch?v=3YmeajE-TT8>
- <https://www.youtube.com/watch?v=Rdd9KAUcvgQ>
- <https://www.youtube.com/watch?v=JaW2jreG8D8>

# ¿Qué es un computador?

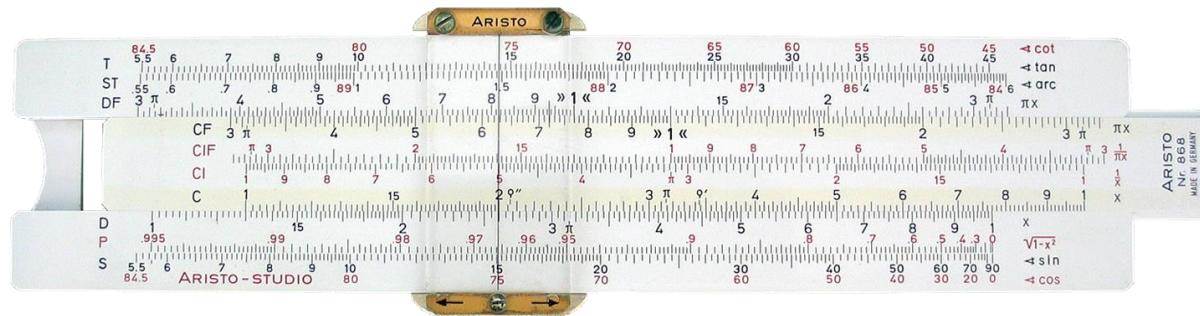
# ... un poquito de historia

La primera calculadora con engranajes.... la pascalina (1623)



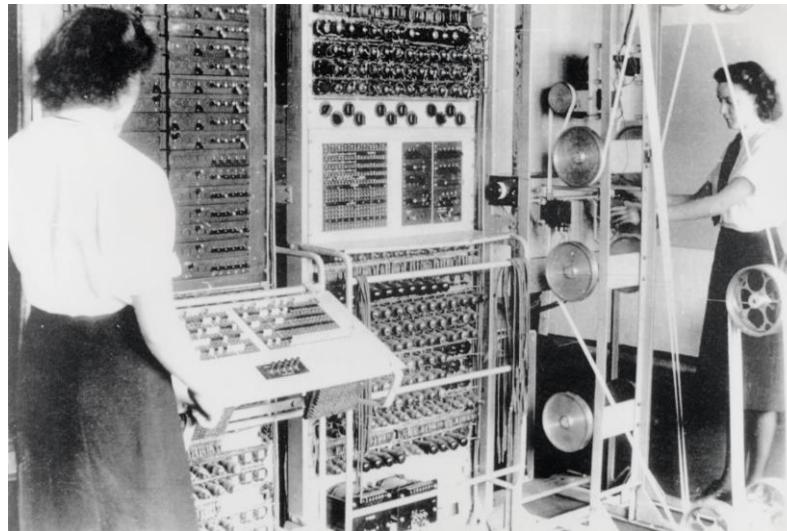
# ... un poquito de historia

Alrededor del siglo XVII se inventó, además de las escalas logarítmicas en la matemática, un instrumento que ayudaba a realizar cálculos aritméticos avanzados: La regla de cálculo (Slide ruler).



# ... un poquito de historia

Principios del siglo XX. Las mujeres eran las computadoras humanas.



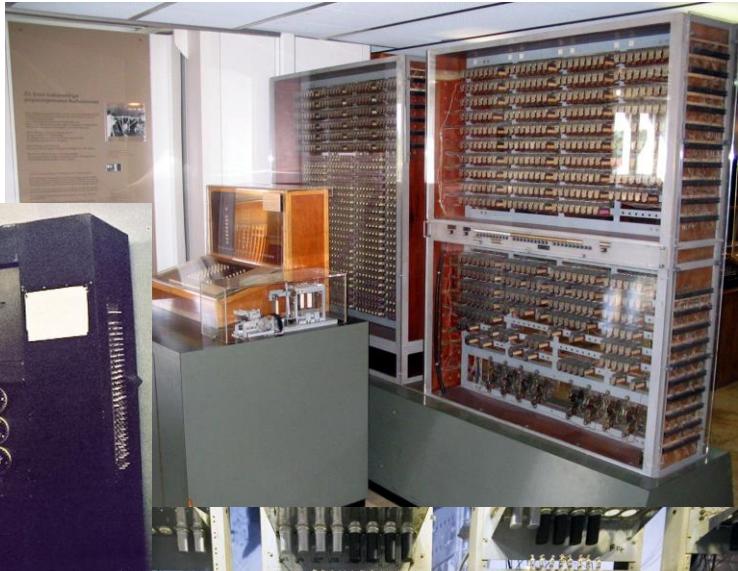
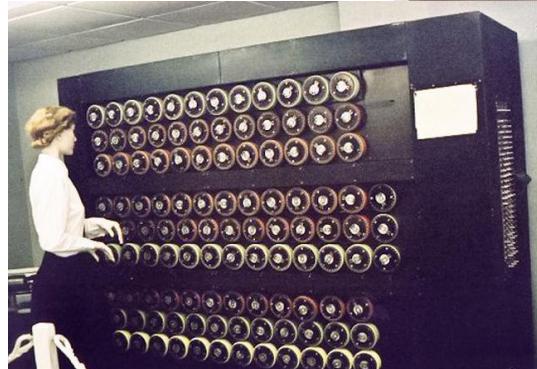
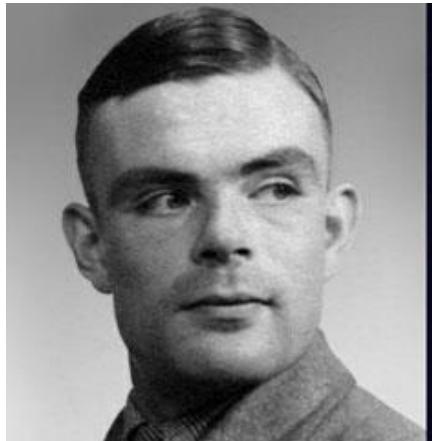
# ... un poquito de historia



Katherine Johnson y Annie Jump Cannon. Dos “computadoras”.

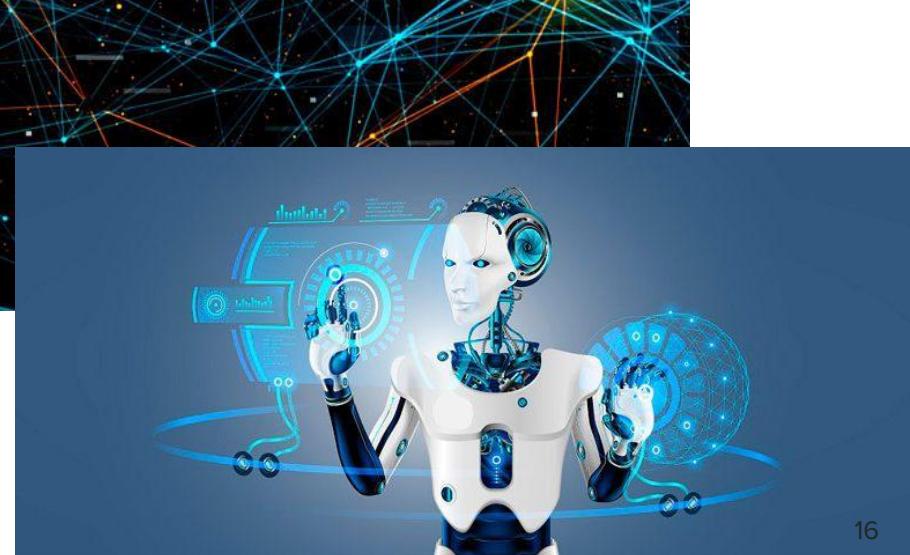
# ... un poquito de historia

A mediados del siglo  
XX.

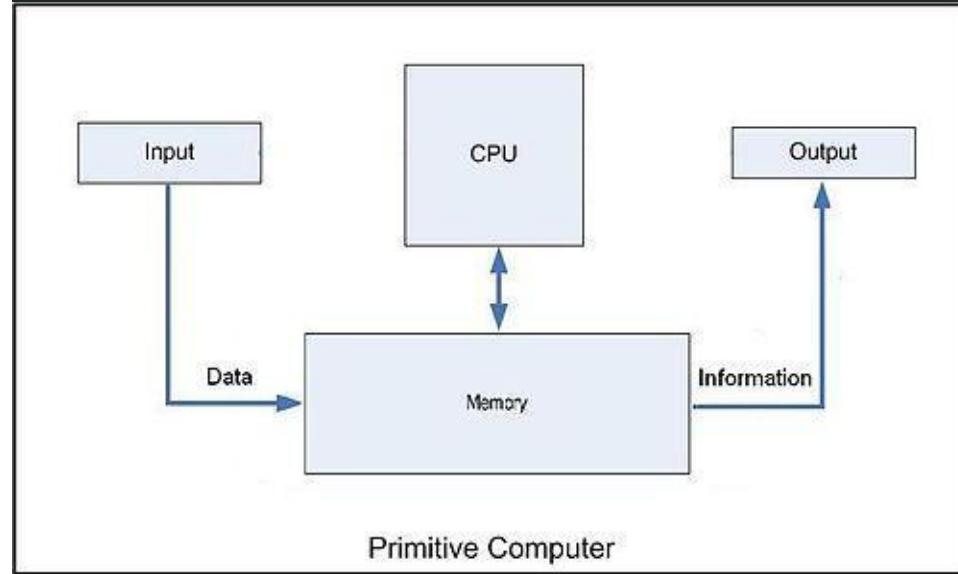


# ... un poquito de historia

¿Hoy cómo estamos?



# Cuáles son las partes más importantes de un computador?



Hay más!

# Tipos de computadores



Los computadores son una herramienta que le permite al hombre realizar actividades automatizadas

# Conclusión

- Los computadores datan desde antes de Cristo
- La gran revolución se dió en el siglo XX
- Los computadores son una herramienta necesaria para la vida tal y como la conocemos hoy en día
- Los computadores están compuestos (principalmente) por dos partes: Hardware y Software

# Bibliografía

- Algoritmia Básica, Roberto Flórez
- C y C++ de afán, Manuel José Páez
- Introducción a la programación con Python, Andrés Marzal.
- Manuales y documentación de referencia en librerías python online.
- Learning Python, Mark Lutz.
- <http://code-reference.com>
- <https://think.cs.vt.edu/blockpy/blockpy/load>
- <http://openbookproject.net/thinkcs/python/english3e/>
- How to Model It: Problem Solving for the Computer Age, A. M. Starfield, K. A. Smith, A. M. Bleloch, Burgess Intl Group.

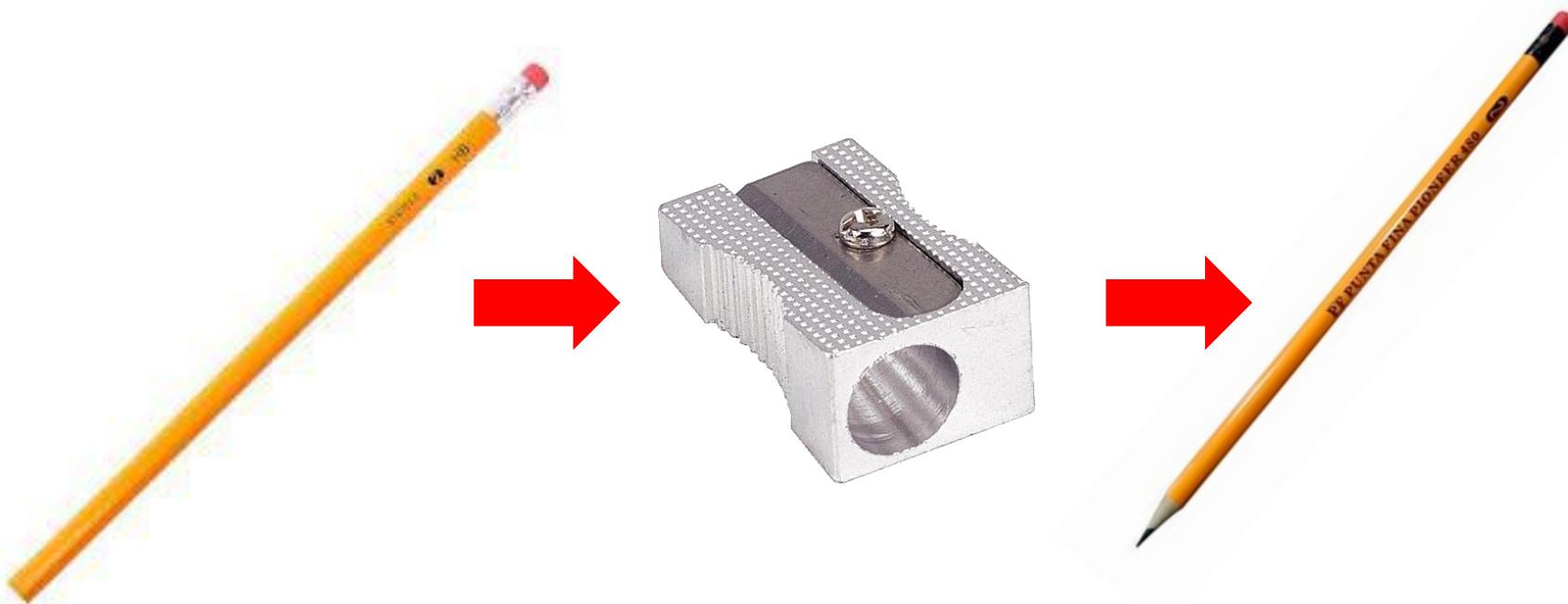
# El Computador



1 8 0 3

# ¿Qué es un computador?

Un computador es en esencia una herramienta que sirve para operar la conversión de una materia prima en un producto, mediante un dispositivo.



# ¿Qué es un computador?

Un computador es en esencia una herramienta que sirve para procesar información.

	A	B	C	D	E	F	G	H	I	J	K
1	colección #	Fecha de muestra	Nombre del sitio	Sitio visitado antes?	Vista	Método de captura	Especie de pescado	Número capturado y liberado	Número capturado y preservado	Número con llagas y deformidades	Nombre archivo de etiqueta fotográfica
2	JR071813-001	19 Jul 2013	Río Hoga Arriba	Si	1	Electroescadora	Gobionellus dormitor	3	0	0	DSC100457.jpg
3	JR071812-001	19 Jul 2013	Río Hoga Arriba	Si	1	Electroescadora	Brycon chagrensis	12	0	1	DSC100453.jpg
4	JR071813-001	19 Jul 2013	Río Hoga Arriba	Si	1	Electroescadora	Ageneiosomus monticola	2	1	0	DSC100459.jpg
5	JR071913-001	19 Jul 2013	Río Hoga Arriba	Si	1	Red de bloques	Gobiomorus dormitor	19	0	0	nada
6	JR071913-001	19 Jul 2013	Río Hoga Arriba	Si	1	Red de bloques	Gephyrocharax intermedius	26	1	1	DSC100460.jpg
7	JR072113-001	21 Jul 2013	Uvero 2	No	1	Electroescadora	Gobiomorus dormitor	2	0	0	DSC100461.jpg
8	JR072113-001	21 Jul 2013	Uvero 2	No	1	Electroescadora	Brycon chagrensis	15	0	0	DSC100462.jpg
9	JR072113-001	21 Jul 2013	Uvero 2	No	1	Electroescadora	Hypseleotris panamensis	8	2	4	DSC100463.jpg
10	JR072113-001	21 Jul 2013	Uvero 2	No	1	Electroescadora	Ageneiosomus monticola	30	0	0	DSC100464.jpg
11	JR072113-001	21 Jul 2013	Uvero 2	No	1	Electroescadora	Aequidens pulcherrimus	1	0	0	DSC100465.jpg
12	JR072113-001	21 Jul 2013	Uvero 2	No	1	Red de bloques	Gobionellus dormitor	7	0	0	nada
13	JR072113-001	21 Jul 2013	Uvero 2	No	1	Red de bloques	Brycon chagrensis	27	0	0	nada
14	JR072113-001	21 Jul 2013	Uvero 2	No	1	Red de bloques	Atherinella chegreali	2	1	0	DSC100466.jpg
15											

Figura 4.5. Ejemplo de base de Datos básicos del sitio de muestreo.



Hay que darle contexto a la información. Por sí sola no es útil

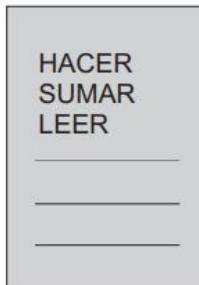
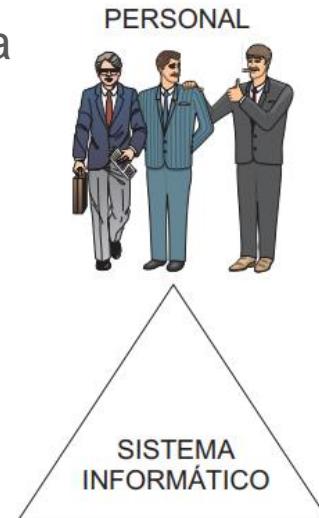
# Sistema Informático

Quien da la instrucción y le da contexto a la información

La herramienta que llevará a cabo la instrucción/operación. Recibe las instrucciones mismas.



HARDWARE



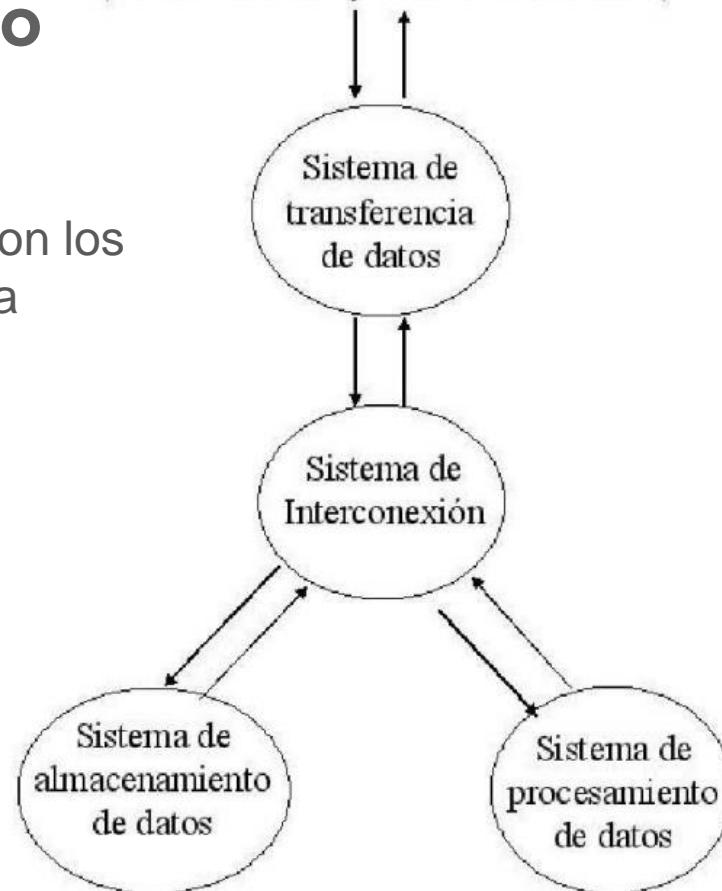
SOFTWARE

Instrucciones y cómo se llevarán a cabo (el programa)

# Funcionamiento

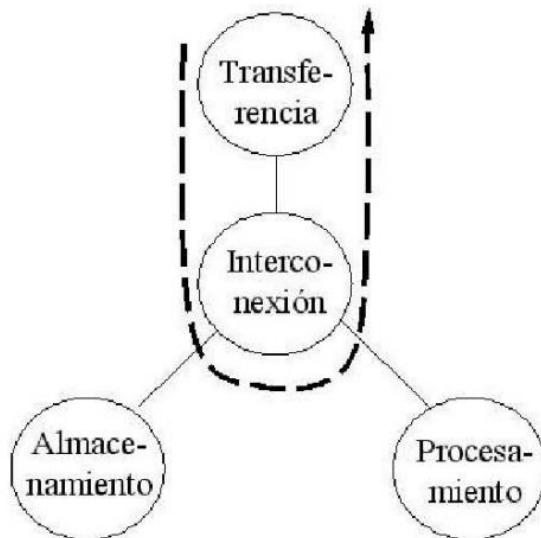
¿Qué se puede hacer con los datos dentro del sistema informático?

(Usuario: Fuente y destino de los datos)

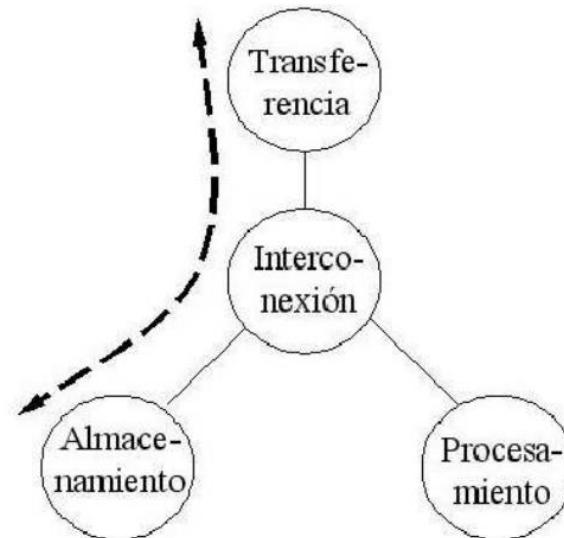


# Funcionamiento

¿Qué se puede hacer con los datos dentro del sistema informático?



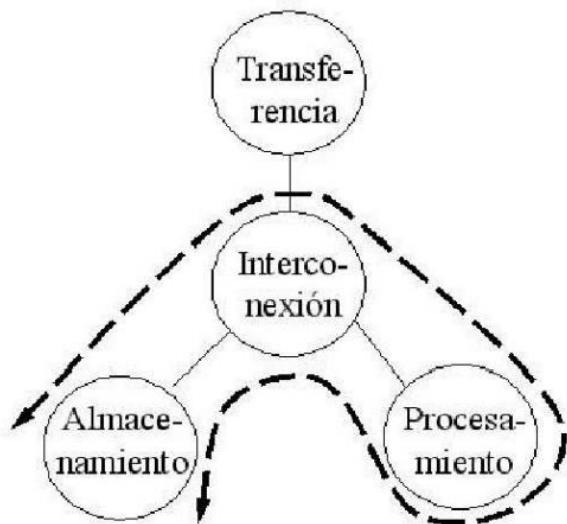
(a) Transferencia de Datos



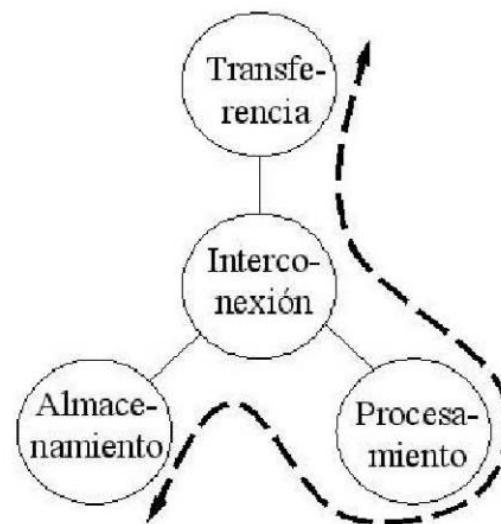
(b) Almacenamiento de Datos

# Funcionamiento

¿Qué se puede hacer con los datos dentro del sistema informático?



(c) Procesamiento de Datos Almacenados



(d) Procesamiento de Datos en Tránsito

# Estructura

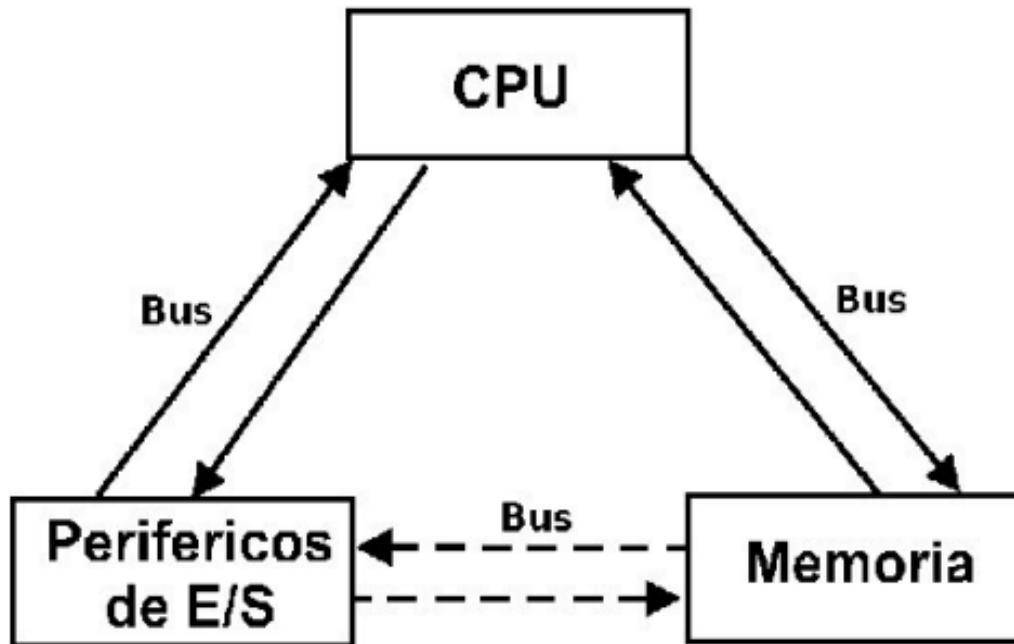
**Unidad Central de Procesamiento:** Ejecución de las instrucciones dadas para el procesamiento de datos. Se divide en unidad de control, aritmético-lógica, interconexión,...

**Unidad de Memoria Principal:** Lugar de dónde serán extraídos los datos para que la CPU sepa cuáles son las instrucciones y donde se almacenarán los datos.

**Unidades de Entrada-Salida (E/S):** Comunicación con el exterior para recibir/exportar los datos antes y luego de ser procesados.

**Sistema de Interconexión:** Mecanismo que proporciona comunicación entre las partes (CPU, memoria y dispositivos E/S).

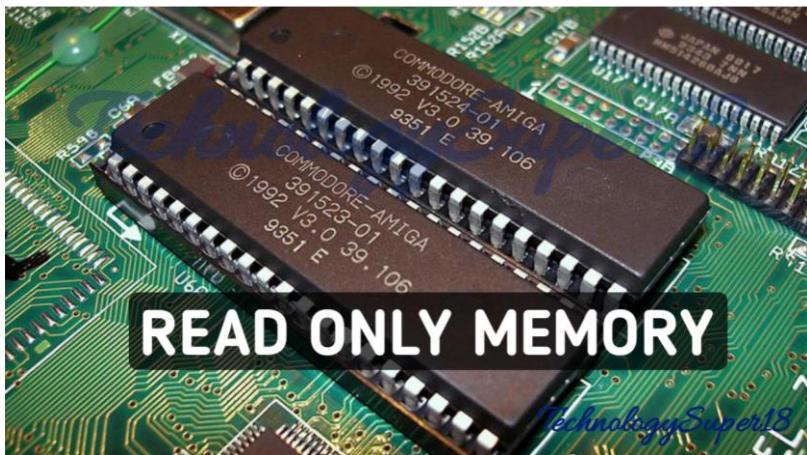
# Arquitectura Básica



# Almacenamiento

Cómo se guarda información en un computador?

La memoria **ROM** guarda información que NO cambia. La memoria contenida en ésta no puede ser modificada. Almacena, por ejemplo, los programas que se deben iniciar al prender el computador.



Memoria **RAM** (Random Access Memory)



La memoria RAM almacena temporalmente información que puede ser modificada en cualquier momento. La información en la memoria RAM se pierde cuando se apaga el computador.

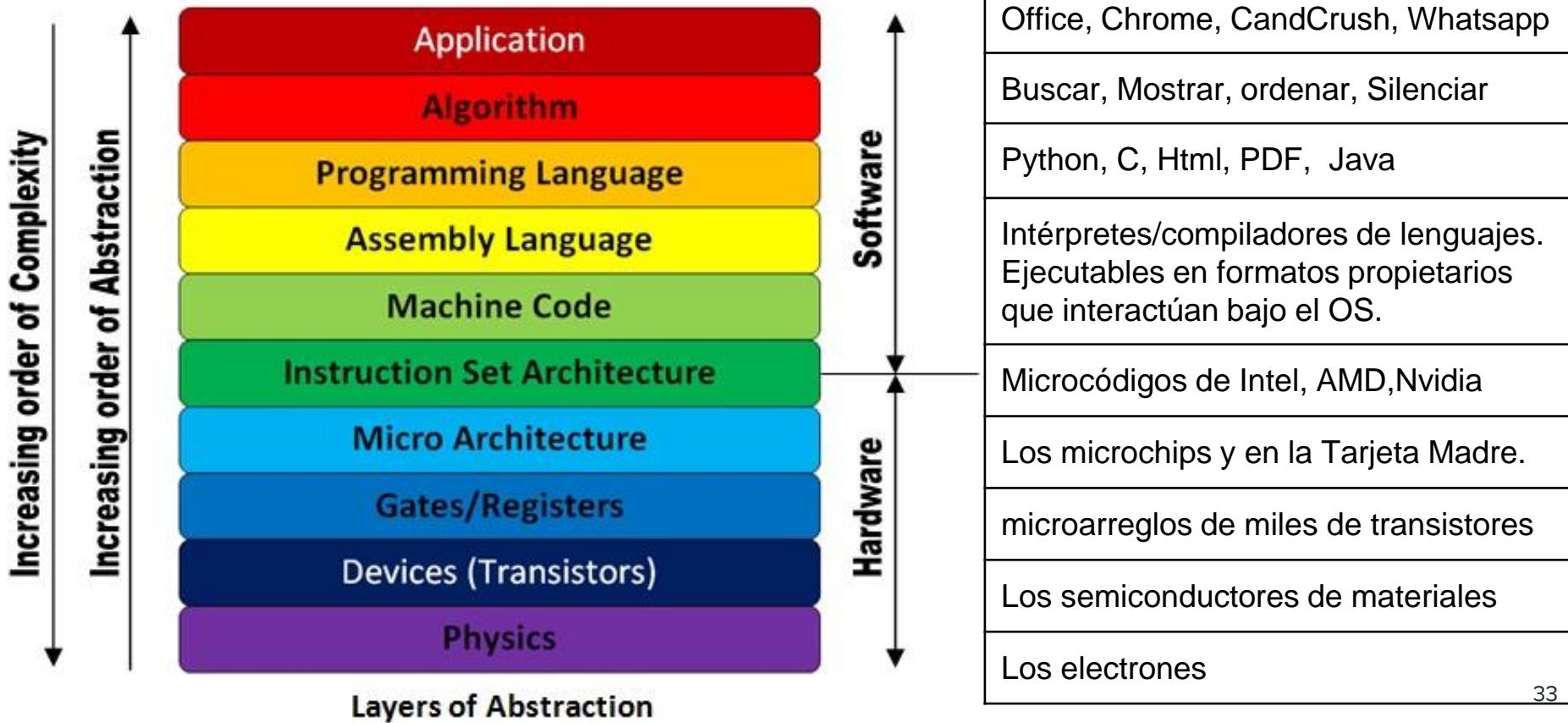
# Almacenamiento

Cómo se guarda información en un computador?



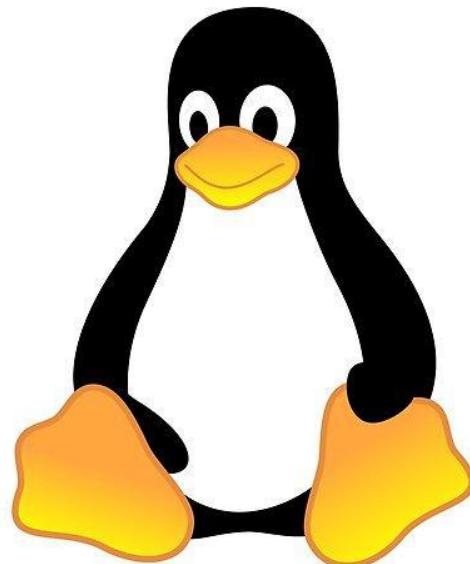
# Partes y almacenamiento

Capas de Abstracción de un computador



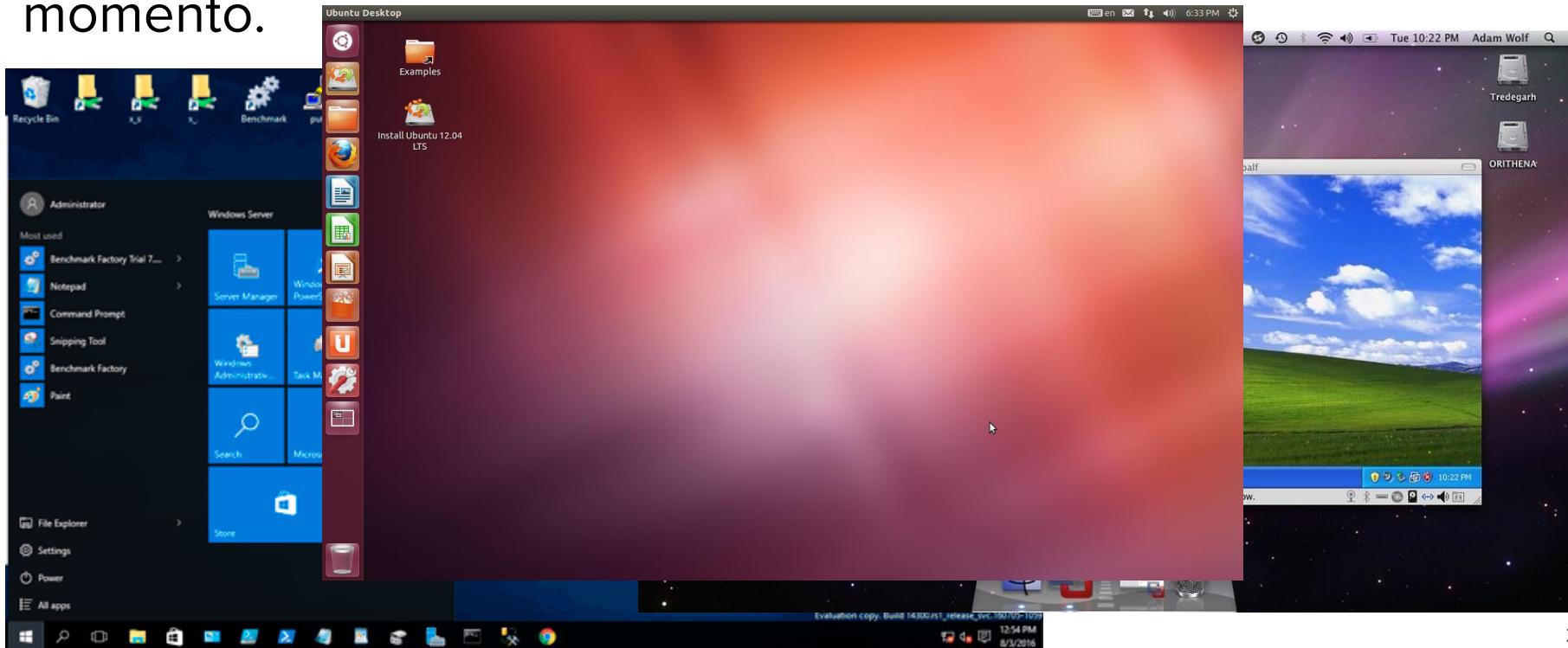
# Sistemas Operativos más conocidos

Éstos son los tres sistemas operativos más usados en este momento.



# Sistemas Operativos más conocidos

Éstos son los tres sistemas operativos más usados en este momento.



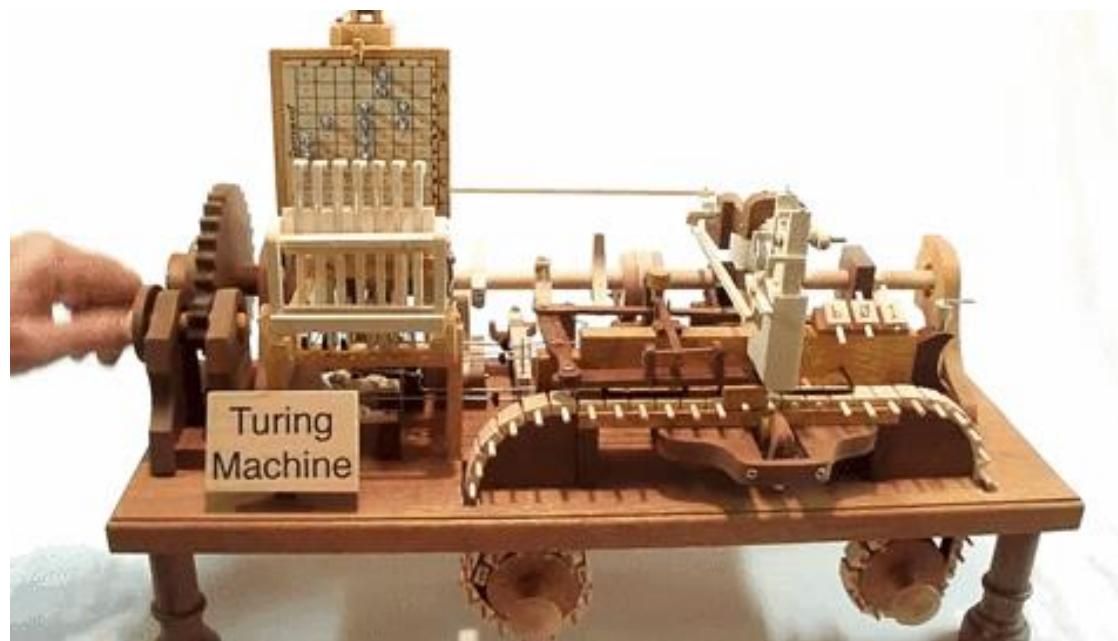
# Conclusión

- Los computadores reciben y entregan información.
- Los sistemas de entrada y salida dependerán del propósito que se tenga.
- El bus de datos es la vía por la que circula la información.
- Hay 3 sistemas operativos principales: Windows, Mac y Linux.

¿Qué es lo que de verdad se guarda en la memoria de un computador?

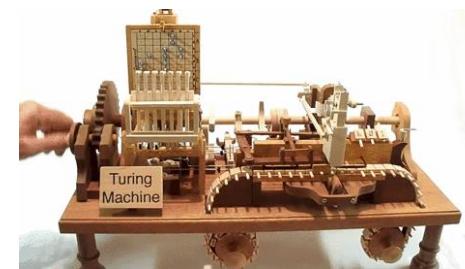
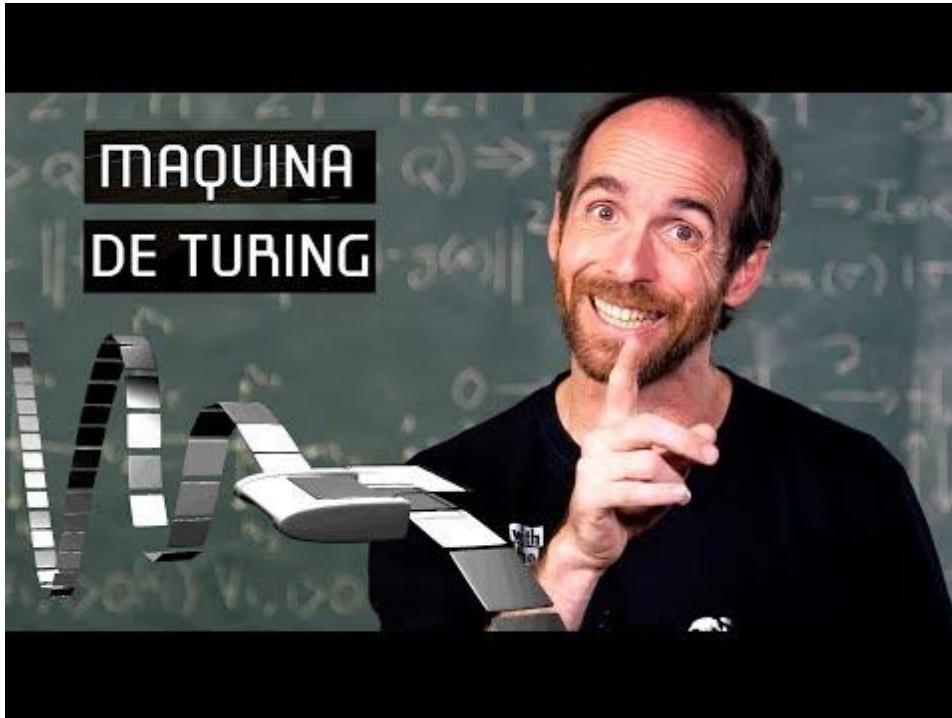
# La Máquina de Turing

Un computador es un aparato que sigue un conjunto ordenado, coherente y recurrente de órdenes.



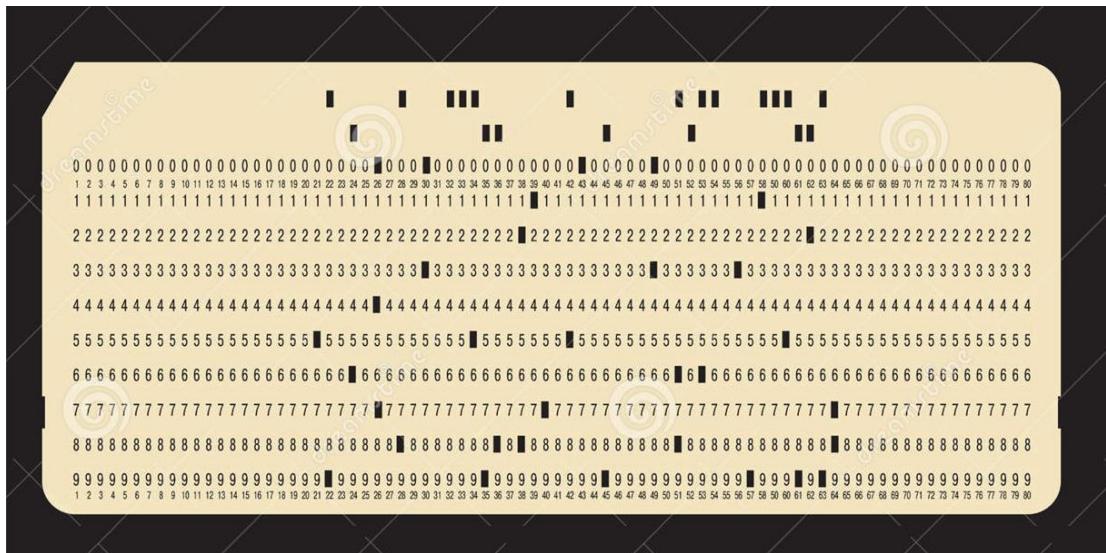
# La Máquina de Turing

Un computador es un aparato que sigue un conjunto ordenado, coherente y recurrente de órdenes.



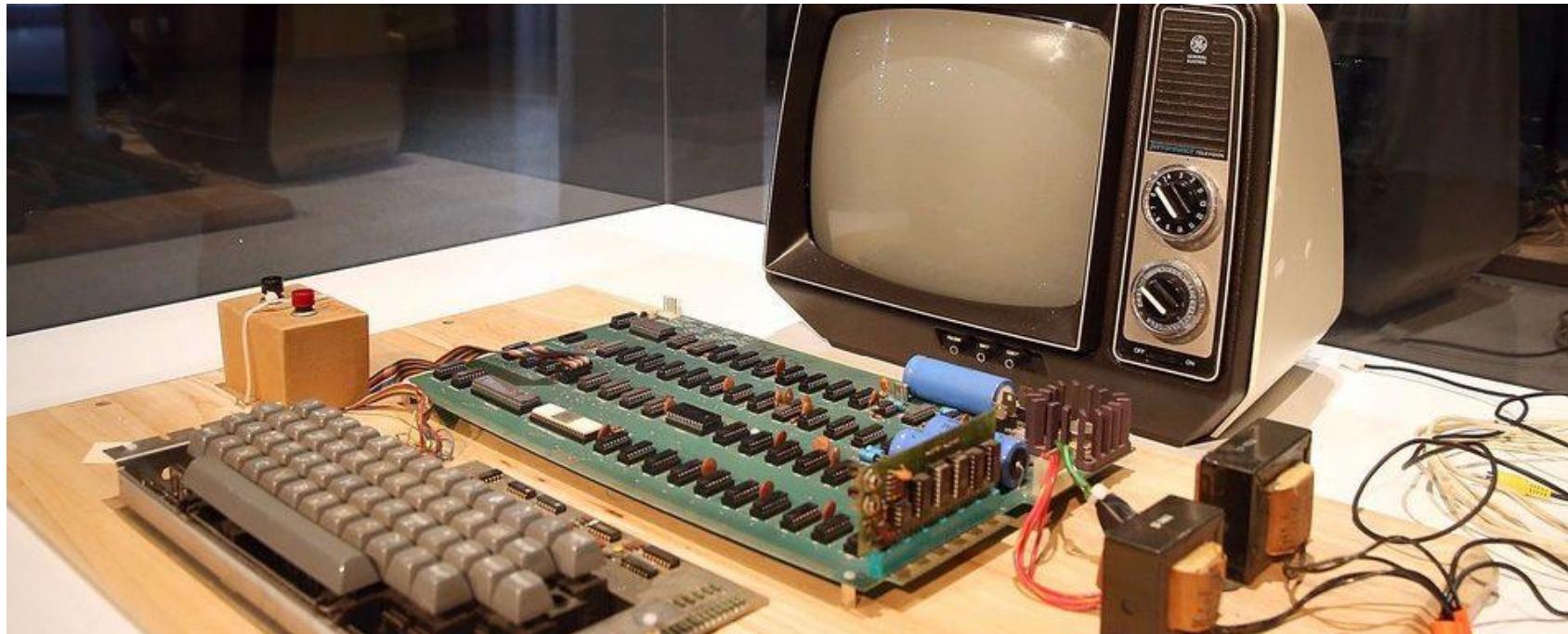
# La Máquina de Turing

Un computador es un aparato que sigue un conjunto ordenado, coherente y recurrente de órdenes.



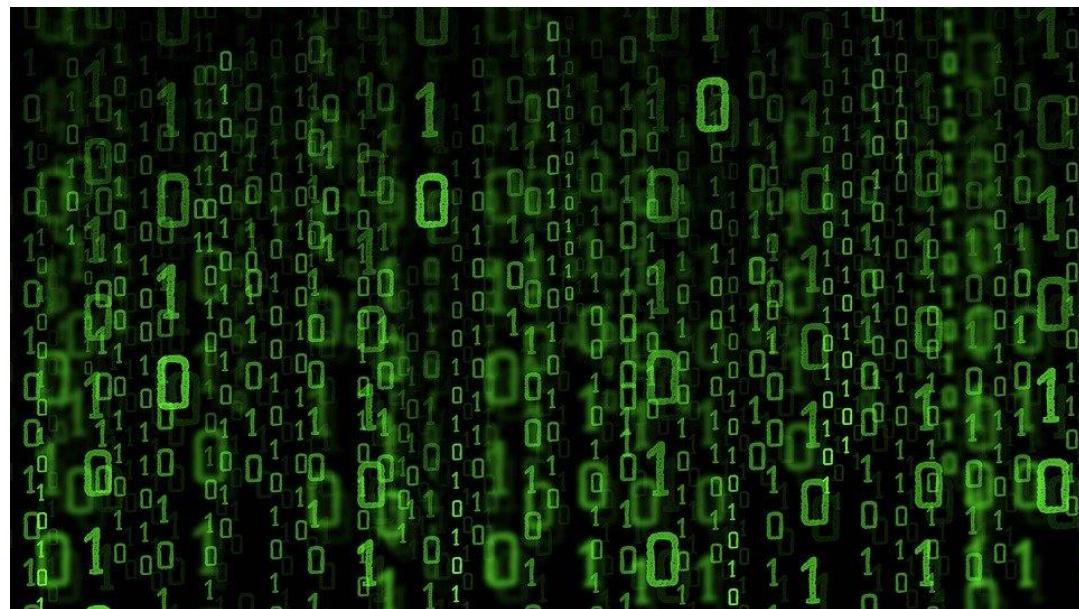
# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0”  
y unos “1”.



# El sistema Binario y Decimal

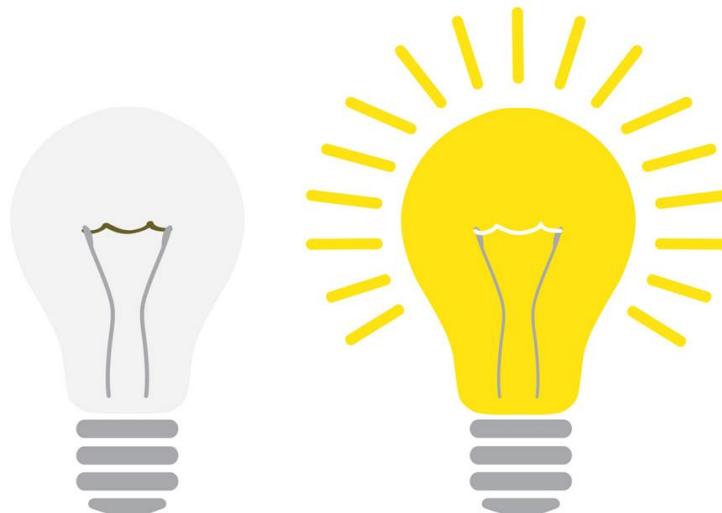
Un computador lo único que entiende son ceros “0” y unos “1”.



Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.



0

1

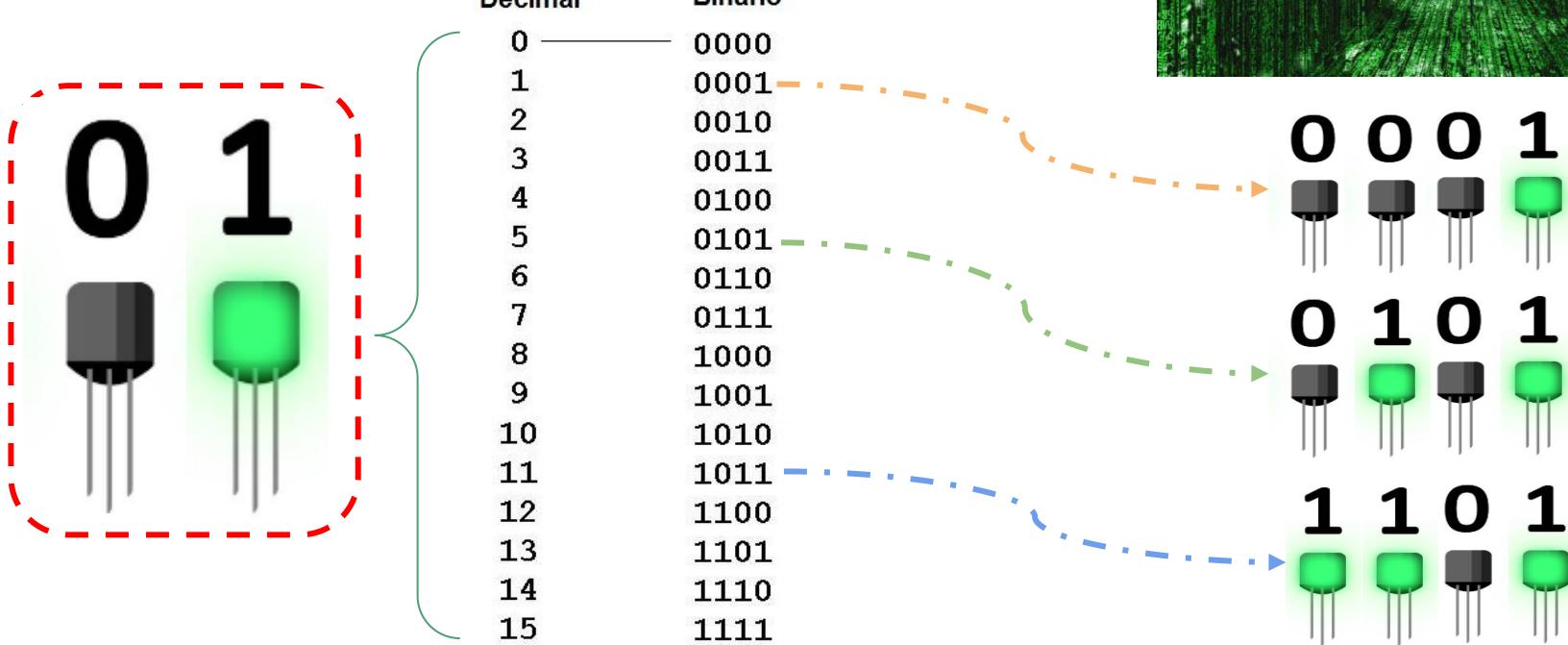
# Binary!

1 0 1 0 1 1

A row of six electronic components, specifically NPN transistors, arranged horizontally. Each transistor has three leads: a long base lead, a shorter collector lead, and a shortest emitter lead. Above each transistor is a black numerical digit representing its state: 1, 0, 1, 0, 1, 1. The second and fifth transistors have their collector leads (the middle leads) glowing green, while the others are dark gray. This visualizes how binary data is stored and processed at the most basic level of electronics.

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.



# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0”  
y unos “1”.



Cuántos números puedo tener a través de combinaciones de bits (o bytes)?

$$2^0=1$$

0

$$2^1=2$$

0, 1

$$2^2=4$$

00, 01, 10, 11

$$2^3=8$$

$$2^4=16$$

$$2^5=32$$

$$2^6=64$$

$$2^7=128$$

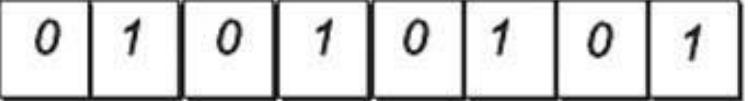
$$2^8=256$$

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.



Esto es un byte



Nombre	Símbolo	Potencias binarias y valores decimales
byte	b	$2^0 = 1$
Kbyte	KB	$2^{10} = 1\,024$
Megabyte	MB	$2^{20} = 1\,048\,576$
Gigabyte	GB	$2^{30} = 1\,073\,741\,824$
Terabyte	TB	$2^{40} = 1\,099\,511\,627\,776$
Petabyte	PB	$2^{50} = 1\,125\,899\,906\,842\,624$
Exabyte	EB	$2^{60} = 1\,152\,921\,504\,606\,846\,976$
Zettabyte	ZB	$2^{70} = 1\,180\,591\,620\,717\,411\,303\,424$
Yottabyte	YB	$2^{80} = 1\,208\,925\,819\,614\,629\,174\,706\,176$

UNIDAD	ABREVIACIÓN	EQUIVALENCIA
bit	b	Dígito binario (1 o 0)
byte	B	8 bits
kilobyte	KB	1,024 bytes o $10^3$ bytes
megabyte	MB	1,024 KB o $10^6$ bytes
gigabyte	GB	1,024 MB o $10^9$ bytes
terabyte	TB	1,024 GB o $10^{12}$ bytes

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.

## Numeric Primitive Data

- “Objects” of different numeric data types occupy different number of cells

Type	Storage	Min Value	Max Value
byte	8 bits	-128	127
short	16 bits	-32,768	32,767
int	32 bits	-2,147,483,648	2,147,483,647
long	64 bits	$< -9 \times 10^{18}$	$> 9 \times 10^{18}$
float	32 bits	$\pm 3.4 \times 10^{38}$ with 7 significant digits	
double	64 bits	$\pm 1.7 \times 10^{308}$ with 15 significant digits	

Rango representable con n bits

- sin signo :  $[2^n - 1 .. 0] \rightarrow \text{total} = 2^n$
- signo-magnitud :  $[-(2^{n-1} - 1) .. (2^{n-1} - 1)] \rightarrow \text{total} = 2^n$

IEEE 754  
format

# El sistema Binario y Decimal

## Números Binarios con signo (Rango)

$$\begin{array}{l} 8 \text{ bits} = 1 \text{ byte} \rightarrow 2^8 = 256 \\ 16 \text{ bits} = 2 \text{ bytes} \rightarrow 2^{16} = 65,536 \\ 32 \text{ bits} = 4 \text{ bytes} \rightarrow 2^{32} = 4.295_{\times 10^9} \end{array} \left. \begin{array}{l} \\ \\ \end{array} \right\} 2^n \xrightarrow{\# \text{ bits}}$$

- Para números con signo:

$$\text{Rango} = -(2^{n-1}) \text{ hasta } +(2^{n-1} - 1)$$

- Con 4 bits:

$$\text{Rango} = -(2^{4-1}) \text{ hasta } +(2^{4-1} - 1)$$

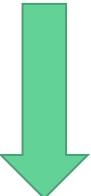
$$= -(2^3) \text{ hasta } +(2^3 - 1)$$

$$= -8 \text{ hasta } +7$$

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.

125



$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
-------	-------	-------	-------	-------	-------	-------

2	125	
2	62	→ 1
2	31	→ 0
2	15	→ 1
2	7	→ 1
2	3	→ 1
2	1	→ 1
	0	→ 1

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0”  
y unos “1”.

125

2	125	
2	62	→ 1
2	31	→ 0
2	15	→ 1
2	7	→ 1
2	3	→ 1
2	1	→ 1
0	0	→ 1

10

2	10	
2	5	→ 0
2	2	→ 1
2	1	→ 0
0	0	→ 1

20

2	20	0
2	10	0
2	5	1
2	2	0
2	1	1

# El sistema Binario y Decimal

## Operación aritméticas Básicas con binarios

SUMA	RESTA	MULTIPL	DIVISIÓN
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$	$0/0 = \text{ind}$
$0 + 1 = 1$	$0 - 1 = 1$ (-1 presta)	$0 * 1 = 0$	$0/1 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$1 * 0 = 0$	$1/0 = \text{ind}$
$1 + 1 = 0$ (+1 lleva)	$1 - 1 = 0$	$1 * 1 = 1$	$1/1 = 1$

Explicaciones y ejemplos de operaciones con binarios (incluye todos los casos) ->

[https://www.youtube.com/watch?v=tdXgB9BENDM&list=PL46-B5QR6sHk5Nck\\_jVP0oJLK79sl-Nhl&index=1](https://www.youtube.com/watch?v=tdXgB9BENDM&list=PL46-B5QR6sHk5Nck_jVP0oJLK79sl-Nhl&index=1)

# Ejemplos aritmética binaria sin signo

SUMA

111+

110

-----

1101

RESTA

10001 -

01010

-----

00111

MULT

10001 \*

11

-----

10001+

10001

-----

110011

DIVISIÓN

101010

-110

1001

-110

0110

110

000

110

111

# Ejemplos aritmética binaria con signo

Suma:  $7 + 6 = 13$

$$\begin{array}{r} 00000111 \\ + 00000110 \\ \hline \end{array}$$

$$\hline \\ 00001101$$

Resta:  $17 - 10 = 7$

$$17 \rightarrow 00010001$$

$$10 \rightarrow 00001010$$

$$-10 \rightarrow 11110110 \text{ (complemento a 2 de 10)}$$

---

**100000000** (Se descarta cuando excede el número de bits)

17-10:

$$00010001+$$

$$11110110$$

---

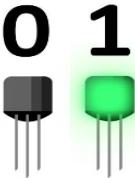
$$\hline \\ \textcolor{red}{100000111} = 7$$

# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0”  
y unos “1”.

¿Y qué pasa con  
números no  
enteros?

# El sistema Binario y Decimal



Un computador lo único que entiende son ceros “0” y unos “1”.

Base 2 (Binary) Place Value Chart to Eight Places: Four Whole Values and Four Fractional Values

Power →

$2^3$	$2^2$	$2^1$	$2^0$
Place Value →	8	4	2
			Ones place
			1

$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$
Fractions are to the right of the radix point.			
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$

Whole numbers are to the left of the radix point.

← Radix Point →

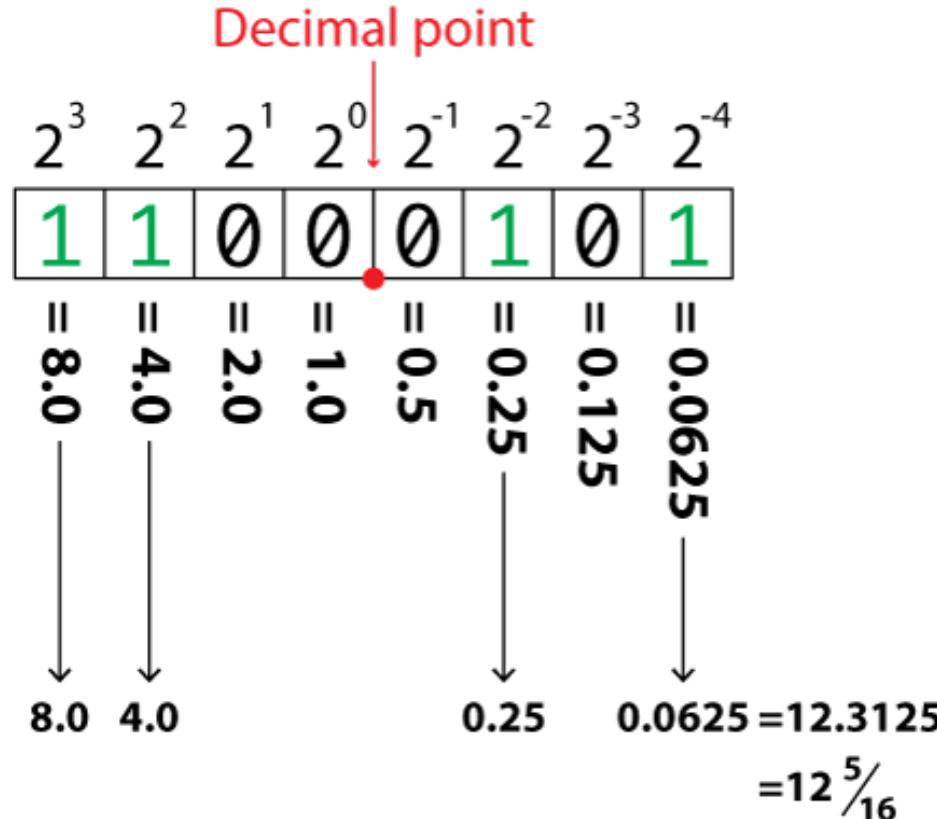
Fractions are to the right of the radix point.



$$8 + 2 + 1 + 0.5 + 0.125 = 11.625$$

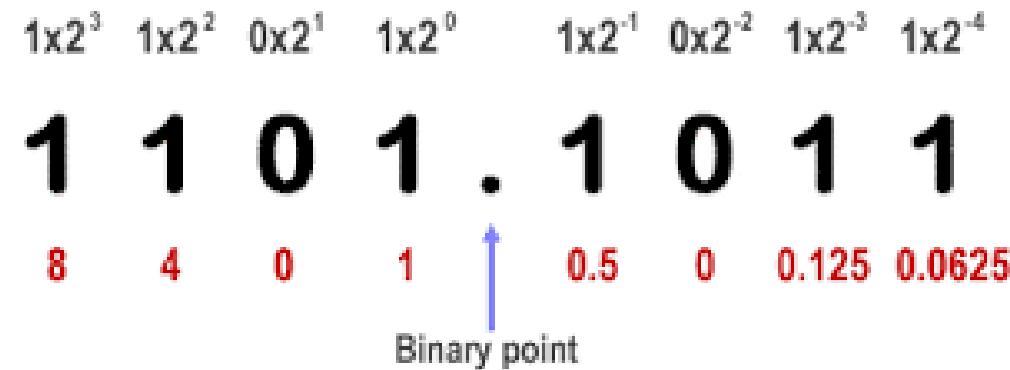
# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.



# El sistema Binario y Decimal

Un computador lo único que entiende son ceros “0” y unos “1”.



$$8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 + 0.0625 = 13.6875 \text{ (Base 10)}$$

# Limitaciones de coma fija (I)

Ejemplo 1: Tomemos 8 posiciones decimales con punto fijo en la mitad.

$$\pi = 0003,1416 \quad \text{en lugar de } 3,1415962$$

se desperdician tres casilla y se usaría mal el espacio disponible para una mejor precisión.

# Limitaciones de coma fija (II)

Ejemplo 2: No se logran acomodar números muy grandes o muy pequeños.

Radio terrestre= 6 371,0000 Km

(Si se necesitara cambiar de unidades a metros no hay 7 espacios en la parte entera para escribirlo! sería 6'731.000,0 )

# FORMATO IEEE 754

Sign Exponent Mantissa

Single Precision (32 bits)



Double Precision (64 bits)



## FORMATO PUNTO FLOTANTE

64bit = precision doble



32bit = precisión simple



bit del signo  
bits del exponente  
bits de la mantisa

16bit = media precisión



# FORMATO IEEE 754

## Cómo escribir número grandes?

Por ejemplo el número 500000 en binario es:

1111010000100100000

$$1111010000100100000 = 1,111\ 010\ 000\ 100\ 100\ 000 * 2^{18}$$

Signo = + = 0

Exponente = 18 + 127 (le sumamos 127 y pasamos a binario) = 145 = 10010001

Mantisa = rellenamos de 0 hasta 23 = 1111 010 000 100 100 000 = 1111 010 000 100 100 000 000 00

500000 = 010010001111010000100100000000000

Excelente explicación en:

<https://www.youtube.com/watch?v=HcjXH9WGmAU>

## Trabajo independiente

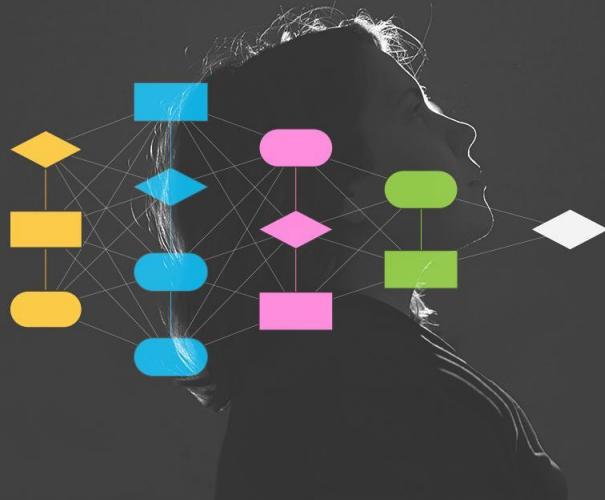
- Estudiar el manejo de números de punto flotante en el documento del profesor Sergio Noriega.

# Conclusión

- El sistema binario y decimal son el lenguaje usado por un computador para interpretar los datos de entrada y salida.
- Un computador es un conjunto de partes que, junto a un sistema operativo, realizan funciones especificadas por el usuario.
- Los sistemas operativos variarán dependiendo de las necesidades del usuario.

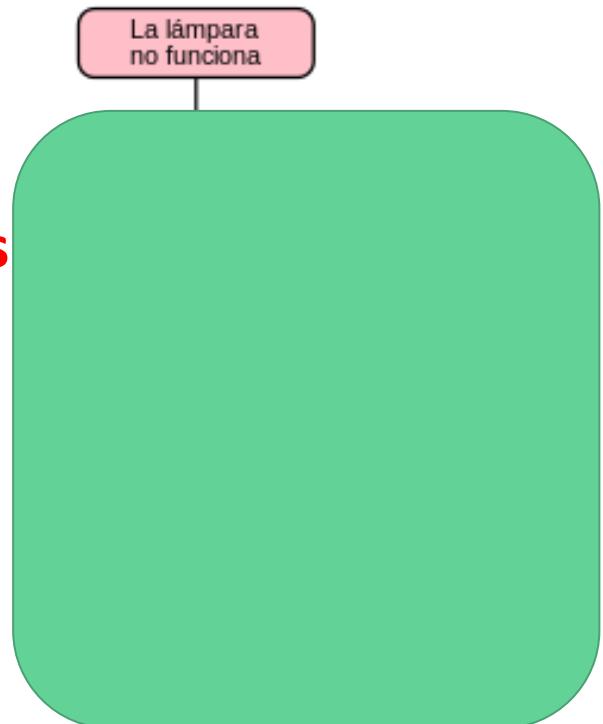
# ¿Qué es la algoritmia?

Primeros pasos para interactuar  
con el computador



# ¿Qué es un algoritmo?

Es un conjunto de instrucciones o reglas **definidas y no-ambiguas, ordenadas y finitas** que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades



¿Qué es un algoritmo?

¿Cómo determinar el área de un triángulo?

# ¿Qué es un algoritmo?

¿Cuántos pasos debo realizar para ir a cine?

1. Realice una lista de pasos a seguir para ir a cine.

# ¿Qué es diagrama de flujo?

Los diagramas de flujo son las descripciones gráficas de los algoritmos.

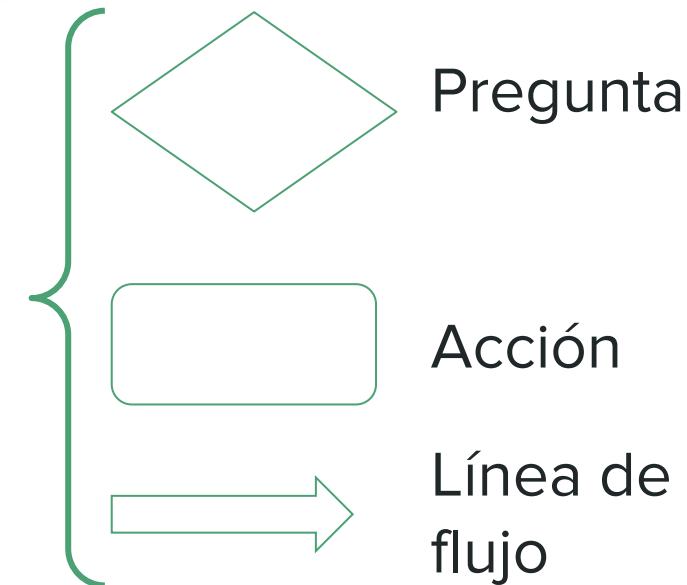
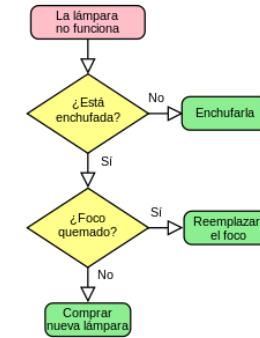
Se construyen a través de símbolos conectados con flechas, indicando la secuencia de instrucciones que debe seguir el algoritmo



# ¿Qué es diagrama de flujo?

## Ejercicio:

1. Escribir un algoritmo para averiguar el nombre, la edad y la estatura de una persona.
2. Usar los símbolos de pregunta y acción para hacer un diagrama de flujo del algoritmo.



Nota: reglas definidas y no-ambiguas, ordenadas y finitas

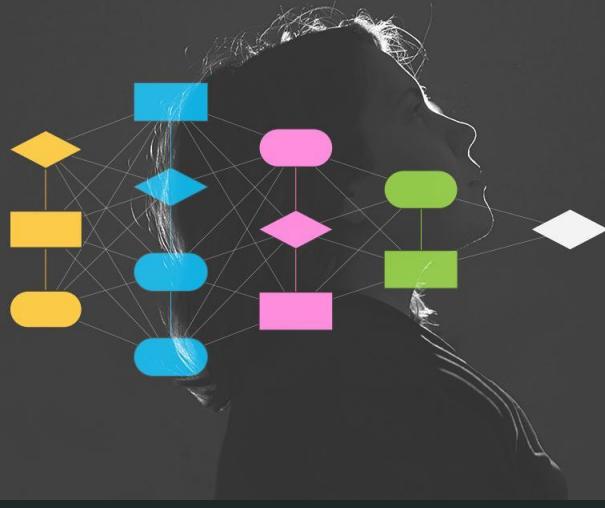
# Conclusión

- Los algoritmos son estructuras que hacen parte fundamental de la programación.
- Los algoritmos son precisos, y el fallo de éstos es un error de quien los genera.
- Se debe ser muy precavido a la hora de construir un algoritmo y tener en cuenta cada paso

# ¿Qué es la algoritmia?

Analizando y ordenando ideas:

1. Comprender
2. Trazar
3. Ejecutar
4. Revisar



# ¿Cómo pensarnos un algoritmo?

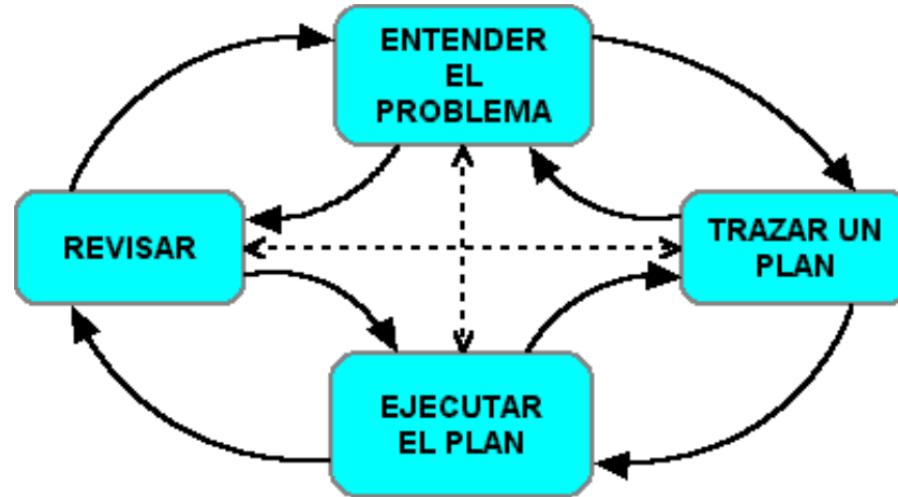
Hay muchos métodos que pueden ayudar atacar un problema y ayudan a llegar a una solución:

- Ensayo y error
- Iluminación
- Lluvia de ideas
- Heurística
- etc.

Para la solución de problemas desde el punto de vista computacional, nos aproximamos al problema desde la **Algoritmia: Aplicar adecuadamente un conjunto de pasos detallados que aseguran una solución correcta**

# ¿Cómo pensarnos un algoritmo?

Para la solución de problemas desde el punto de vista computacional, nos aproximamos al problema desde la **Algoritmia**



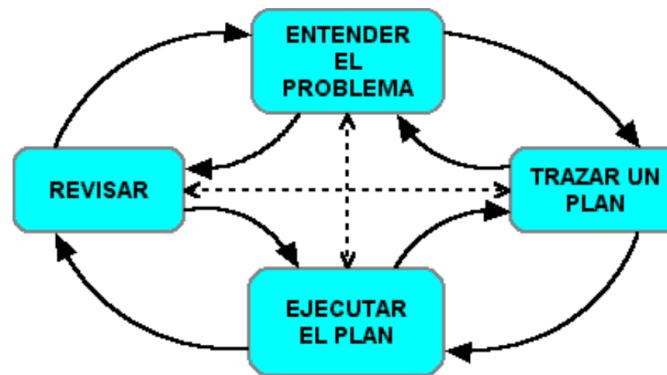
# ¿Cómo pensarnos un algoritmo?

Saber qué queremos hacer

Revisar el resultado

Determinar los pasos a seguir para resolver

Ejecutar los pasos a seguir



# ¿Cómo pensarnos un algoritmo?

## 1. ***COMPRENDER EL PROBLEMA.***

- Leer el problema varias veces
- Establecer los datos del problema
- Aclarar lo que se va a resolver (¿Cuál es la pr
- Precisar el resultado que se desea lograr
- Determinar la incógnita del problema
- Organizar la información
- Agrupar los datos en categorías
- Trazar una figura o diagrama.

## 2. ***HACER EL PLAN.***

- Escoger y decidir las operaciones a efectuar.
- Eliminar los datos inútiles.
- Descomponer el problema en otros más pequeños.

## 3. ***EJECUTAR EL PLAN (Resolver).***

- Ejecutar en detalle cada operación.
- Simplificar antes de calcular.
- Realizar un dibujo o diagrama

## 4. ***ANALIZAR LA SOLUCIÓN (Revisar).***

- Dar una respuesta completa
- Hallar el mismo resultado de otra manera.
- Verificar por apreciación que la respuesta es adecuada.

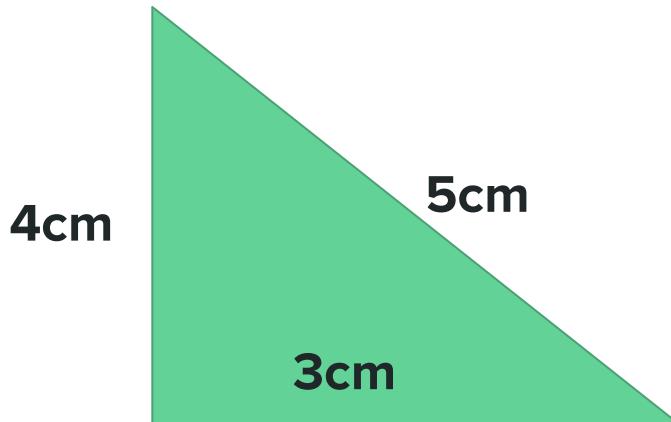
# ¿Cómo pensarnos un algoritmo?

Ejercicio:

Calcular el área de un triángulo rectángulo cuya base mide 3cm, la altura mide 4cm y la hipotenusa mide 5cm

# ¿Cómo pensarnos un algoritmo?

Calcular el área de un triángulo rectángulo cuya base mide 3cm, la altura mide 4cm y la hipotenusa mide 5cm



**Formular el problema:** Ya se encuentra claramente planteado.

**Resultados esperados:** El área de un triángulo rectángulo.

**Datos disponibles:** Base, Altura, Hipotenusa, tipo de triángulo. La incógnita es el área y todos los valores son constantes. El valor de la hipotenusa se puede omitir. El estudiante debe preguntarse si sus conocimientos actuales de matemáticas le permiten resolver este problema; de no ser así, debe plantear una estrategia para obtener los conocimientos requeridos.

**Determinar las restricciones:** Utilizar las medidas dadas.

**Procesos necesarios:** Guardar en dos variables los valores de Base y Altura; Guardar en una constante el divisor 2; aplicar la fórmula  $\text{área}=\text{base}*\text{altura}/2$ ; comunicar el resultado (área).

# ¿Cómo pensarnos un algoritmo?

En un juego, el ganador obtiene una ficha roja; el segundo, una ficha azul; y el tercero, una amarilla. Al final de varias rondas, el puntaje se calcula de la siguiente manera: Al cubo de la cantidad de fichas rojas se adiciona el doble de fichas azules y se descuenta el cuadrado de las fichas amarillas. Si Andrés llegó 3 veces en primer lugar, 4 veces de último y 6 veces de intermedio, ¿Qué puntaje obtuvo? (Adaptado de Melo (2001), página 30).



## COMPRENDE

- Leer detenidamente el problema
- ¿Cuántos colores de fichas se reparten?
- ¿Cuántas fichas rojas, azules y amarillas obtuvo Andrés?
- ¿Qué pregunta el problema?



## 1. COMPRENDER EL PROBLEMA.

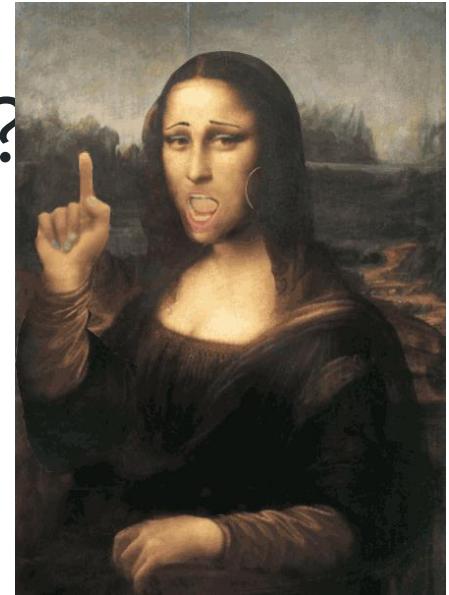
- Leer el problema varias veces
- Establecer los datos del problema
- Aclarar lo que se va a resolver (¿Cuál es la pregunta?)
- Precisar el resultado que se desea lograr
- Determinar la incógnita del problema
- Organizar la información
- Agrupar los datos en categorías
- Trazar una figura o diagrama.

# ¿Cómo pensarnos un algoritmo?

En un juego, el ganador obtiene una ficha roja; el segundo, una ficha azul; y el tercero, una amarilla. Al final de varias rondas, el puntaje se calcula de la siguiente manera: Al cubo de la cantidad de fichas rojas se adiciona el doble de fichas azules y se descuenta el cuadrado de las fichas amarillas. Si Andrés llegó 3 veces en primer lugar, 4 veces de último y 6 veces de intermedio, ¿Qué puntaje obtuvo? (Adaptado de Melo (2001), página 30).

## PLAÑEA

- Para hallar el puntaje que obtiene Andrés por sus llegadas de primero, calcular el cubo de la cantidad de fichas rojas.
- Para hallar el puntaje por sus llegadas en segundo lugar, calcular el doble de la cantidad de fichas azules.
- Para hallar el puntaje que pierde por sus llegadas en último lugar, calcular el cuadrado de la cantidad de fichas amarillas.
- Para hallar el puntaje total, calcular la suma de los puntajes por las fichas rojas y azules, restarle los puntos de las fichas amarillas.



# ¿Cómo pensarnos un algoritmo?

En un juego, el ganador obtiene una ficha roja; el segundo, una ficha azul; y el tercero, una amarilla. Al final de varias rondas, el puntaje se calcula de la siguiente manera: Al cubo de la cantidad de fichas rojas se adiciona el doble de fichas azules y se descuenta el cuadrado de las fichas amarillas. Si Andrés llegó 3 veces en primer lugar, 4 veces de último y 6 veces de intermedio, ¿Qué puntaje obtuvo? (Adaptado de Melo (2001), página 30).



## RESUELVE

- Por tres fichas rojas:  $3^3 = 27$  puntos
- Por seis fichas azules:  $6 \times 2 = 12$  puntos
- Por cuatro fichas amarillas:  $4^2 = 16$  puntos
- Para obtener el puntaje final de Andrés, sumar los puntos obtenidos con las fichas rojas y azules ( $27 + 12 = 39$  puntos) y de este resultado restar los puntos representados por las fichas amarillas ( $39 - 16 = 23$  puntos).



# ¿Cómo pensarnos un algoritmo?

En un juego, el ganador obtiene una ficha roja; el segundo, una ficha azul; y el tercero, una amarilla. Al final de varias rondas, el puntaje se calcula de la siguiente manera: Al cubo de la cantidad de fichas rojas se adiciona el doble de fichas azules y se descuenta el cuadrado de las fichas amarillas. Si Andrés llegó 3 veces en primer lugar, 4 veces de último y 6 veces de intermedio, ¿Qué puntaje obtuvo? (Adaptado de Melo (2001), página 30).



## REVISA

- El puntaje que obtuvo Andrés es 23 puntos.
- Verificar las operaciones y comparar los cálculos con la solución estimada.

# ¿Cómo pensarnos un algoritmo?

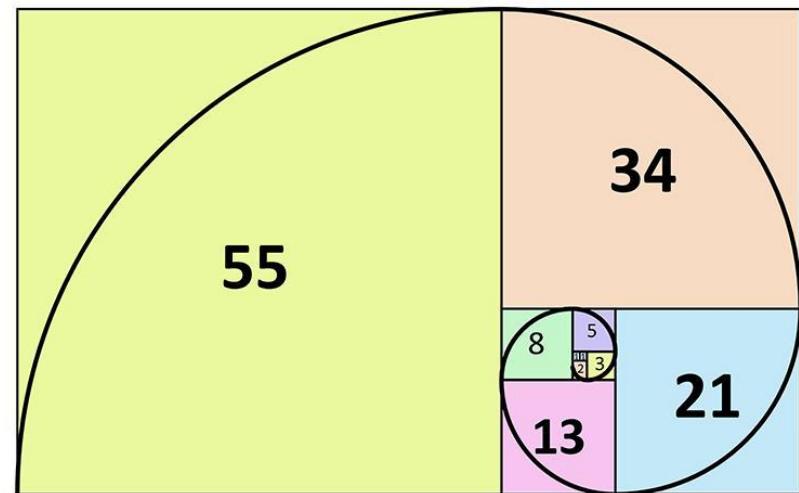
Ejercicio:

Calcular la nota final de un curso que tiene 4 evaluaciones, todas con el mismo porcentaje.

# ¿Cómo pensarnos un algoritmo?

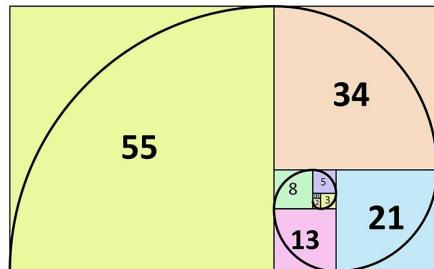
Ejercicio:

Calcular los primeros 100 números de la serie de Fibonacci



# ¿Cómo pensarnos un algoritmo?

Ejercicio:



## The Fibonacci Sequence

1,1,2,3,5,8,13,21,34,55,89,144,233,377...

Calcular los primeros 100 números de la serie de Fibonacci

$$1+1=2$$

$$1+2=3$$

$$2+3=5$$

$$3+5=8$$

$$5+8=13$$

$$8+13=21$$

$$13+21=34$$

$$21+34=55$$

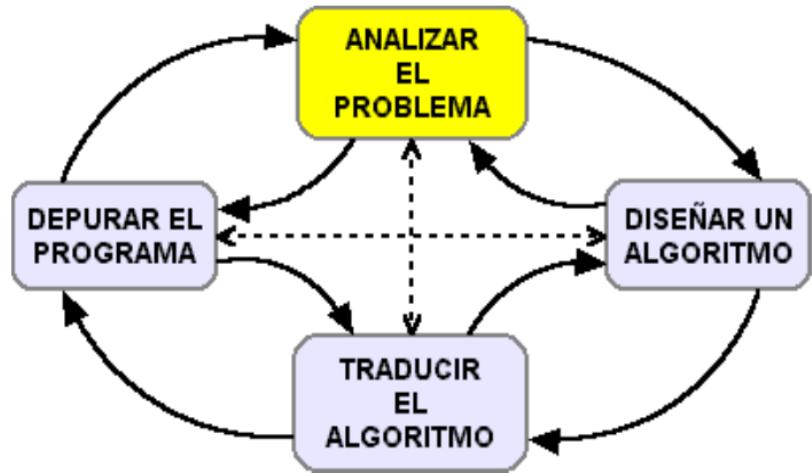
$$34+55=89$$

$$55+89=144$$

$$89+144=233$$

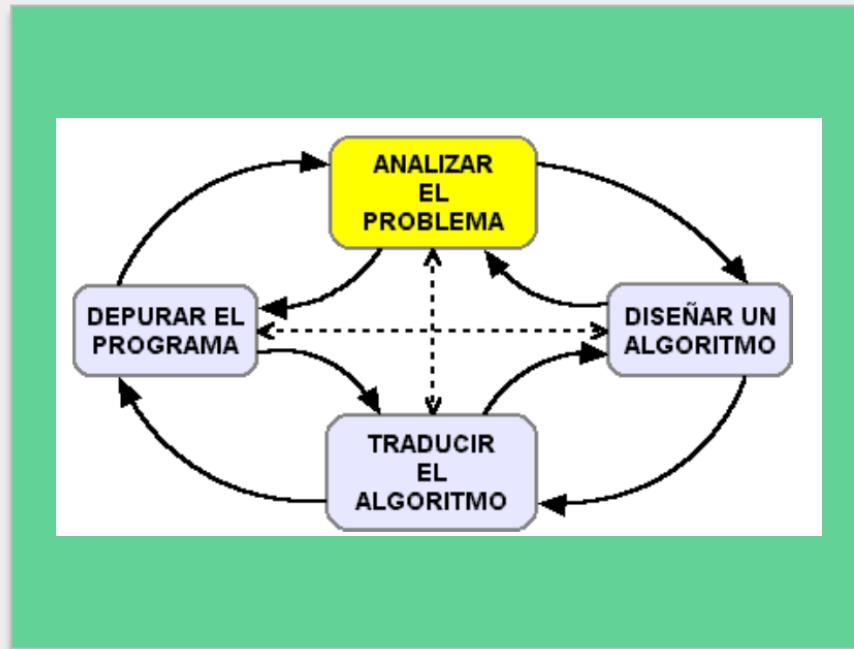
$$144+233=377$$

# ¿Cómo pensarnos un algoritmo?



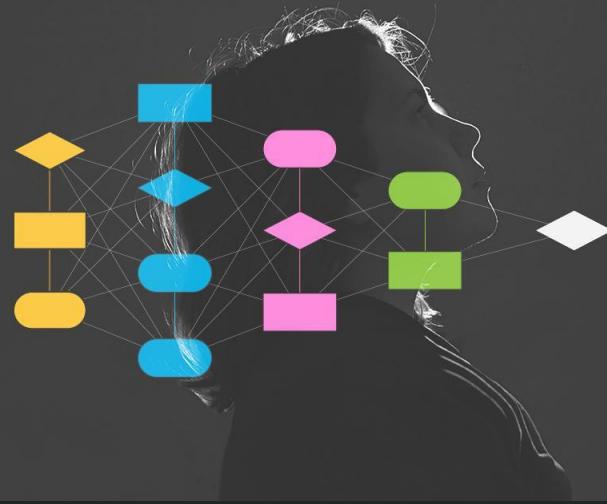
# Conclusiones

- Para la programación se debe ser organizado en la solución de problemas.
- Tratar, en lo posible, de seguir los pasos mínimos propuestos.
- Items importantes: Algoritmo, Pseudo-código, programa, resultado, evaluación



# Programando

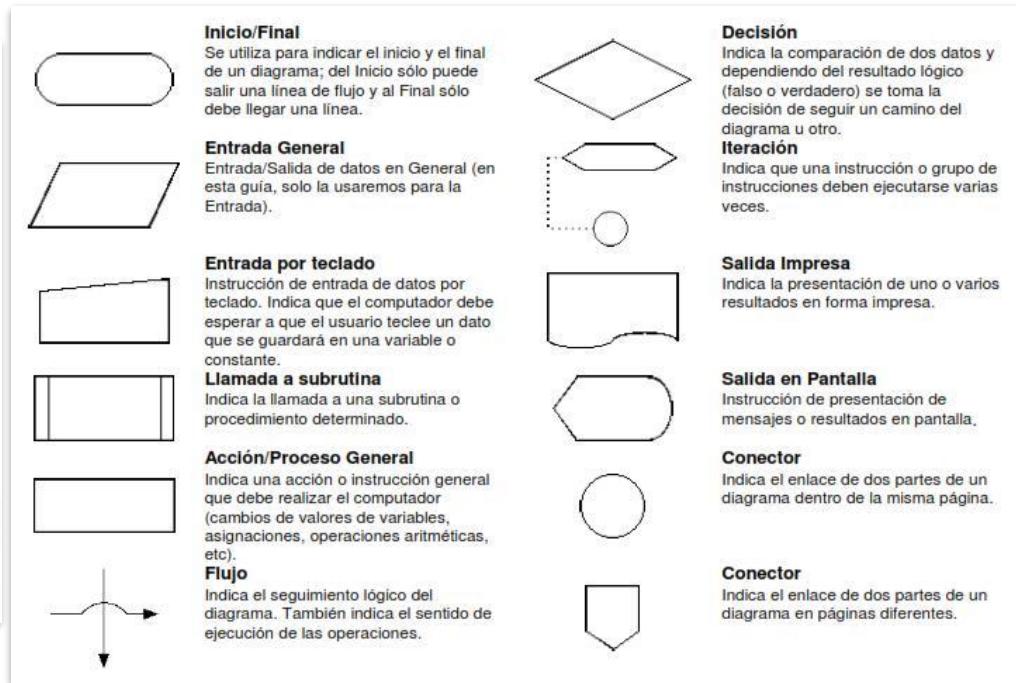
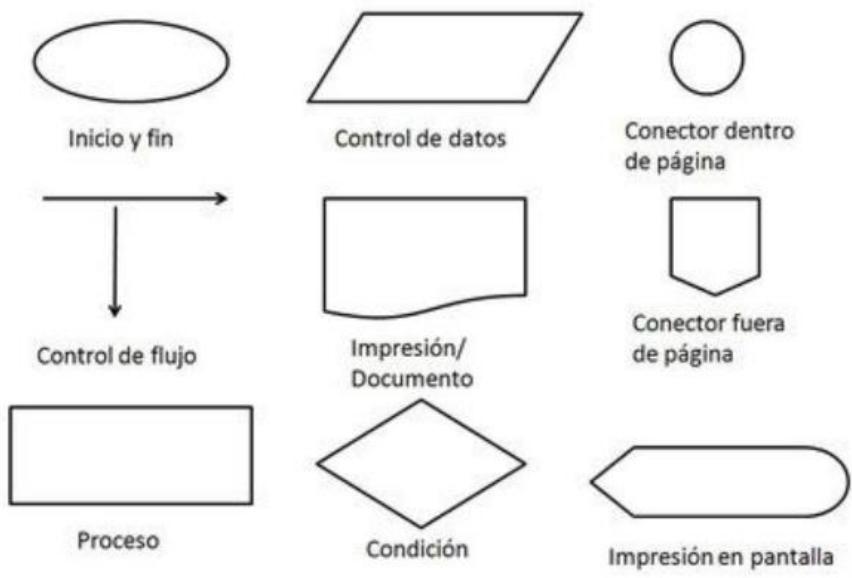
¿Cómo pasar a un código Los  
diagramas de flujo?



# Conjunto básico de órdenes

- **Leer, Entrada:** Estas se usan para recibir información por pantalla o por archivos.
- **Imprimir, Escribir, Salida:** Estas se usan para sacar información en pantalla o por archivos.
- **Si, Si no:** Estás se usan para preguntar, si pasa esto sucede aquello, sino, sucede aquello otro.
- **Para, Mientras:** Estas se usan para los ciclos dependiendo de la longitud de la repetición.

# Diagrama de flujo



# Conjunto básico de órdenes

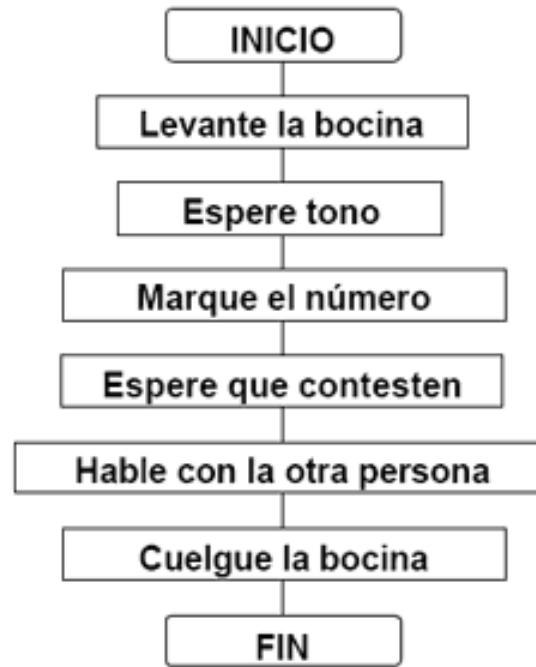
Pseudocódigo:

INICIO

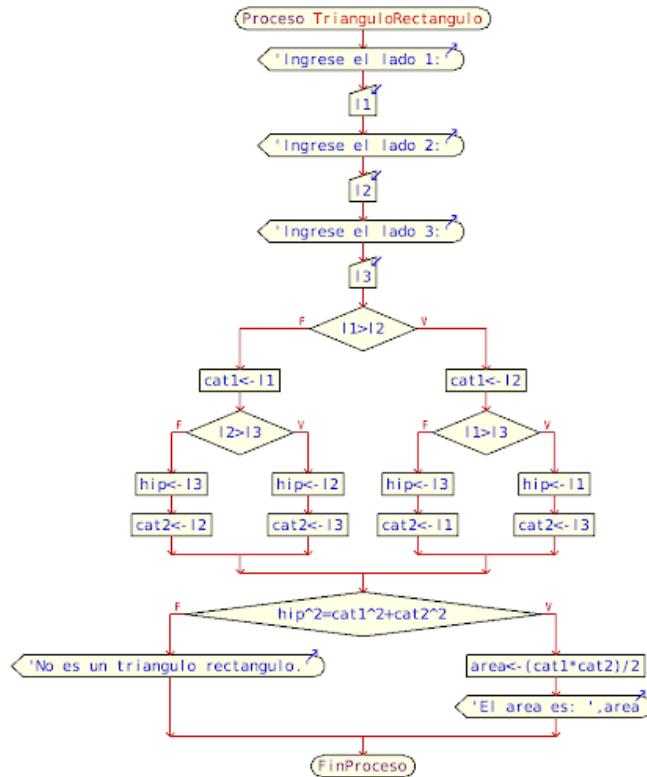
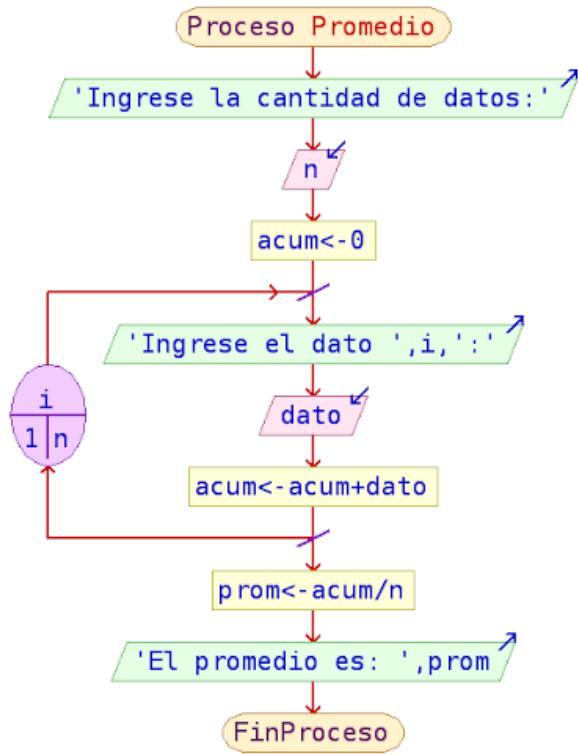
Levante la bocina  
Espere tono  
Marque el número  
Espere que contesten  
Hable con la otra persona  
Cuelgue la bocina

FIN

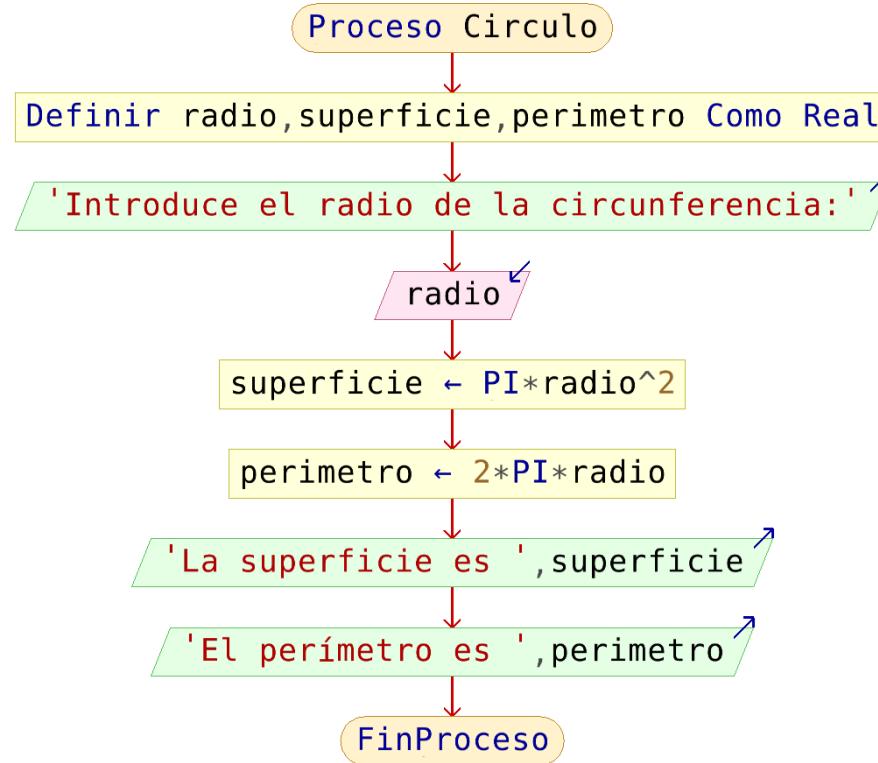
Diagrama de flujos:



# Diagrama de flujo



# Diagrama de flujo





BUCLAS



VARIABLES



CONDICIONALES



# Condicionales y bucles

1. Si
  2. Si no
  3. Para
  4. Mientras
-

# Tipos básicos de datos

## Datos numéricos

estoEsUnNúmero = 98.34



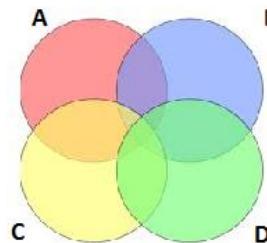
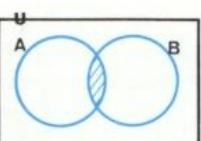
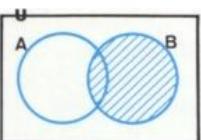
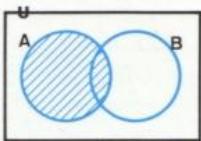
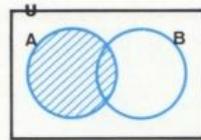
## Datos caráteres

estoEsUnConjuntoDeLetras="Una\_cadena"

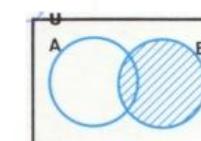
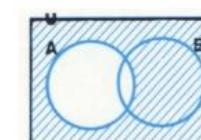
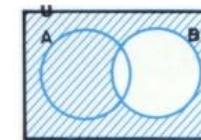
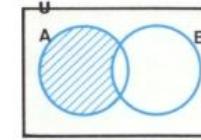
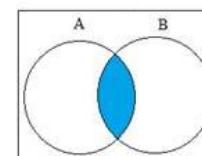
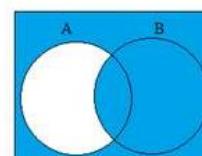
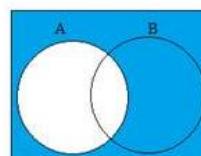
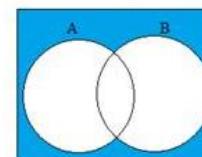
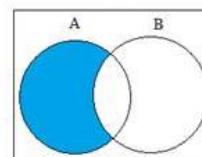
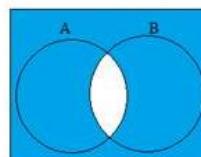
## Datos Booleanos

estoEsUnaBooleana = True

# Operadores lógicos



Operaciones entre conjuntos



B

# Operadores lógicos

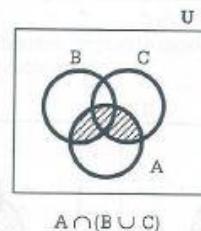
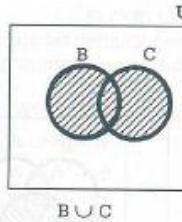
Tablas de verdad					
P	Q	P ^ Q AND Conjunción	P v Q OR Disyunción	~P NOT Negación	~Q NOT Negación
V	V	V	V	F	F
V	F	F	V	F	V
F	F	F	V	V	V
F	F	F	F	V	V

Operador	Significado	True	False
&&	Operador AND conjunción	(7<9)&&(7<15)	(-100>0)&&(-100<500)
	Operador OR disyunción	(5>2)    (5<3)	(5<2)    (5<0)
!	Operador NOT Negación	!(5<3)	!(7<11)

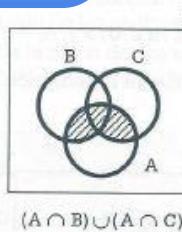
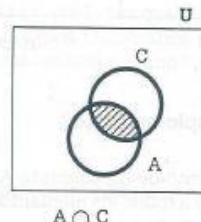
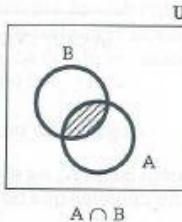
# Operadores lógicos

Los siguientes diagramas de Venn ilustran la validez de esta ley:

Unión



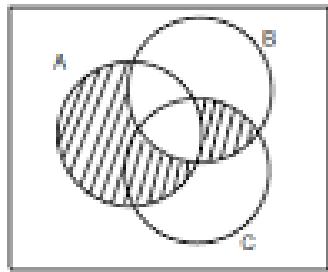
Intersección



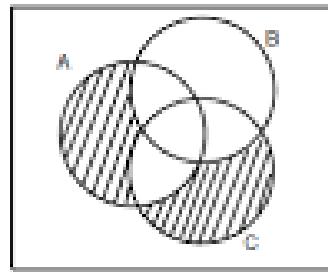
# Operadores lógicos

Hacerlos!

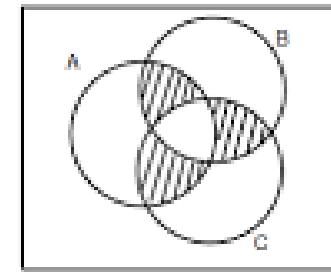
i)



ii)



iii)



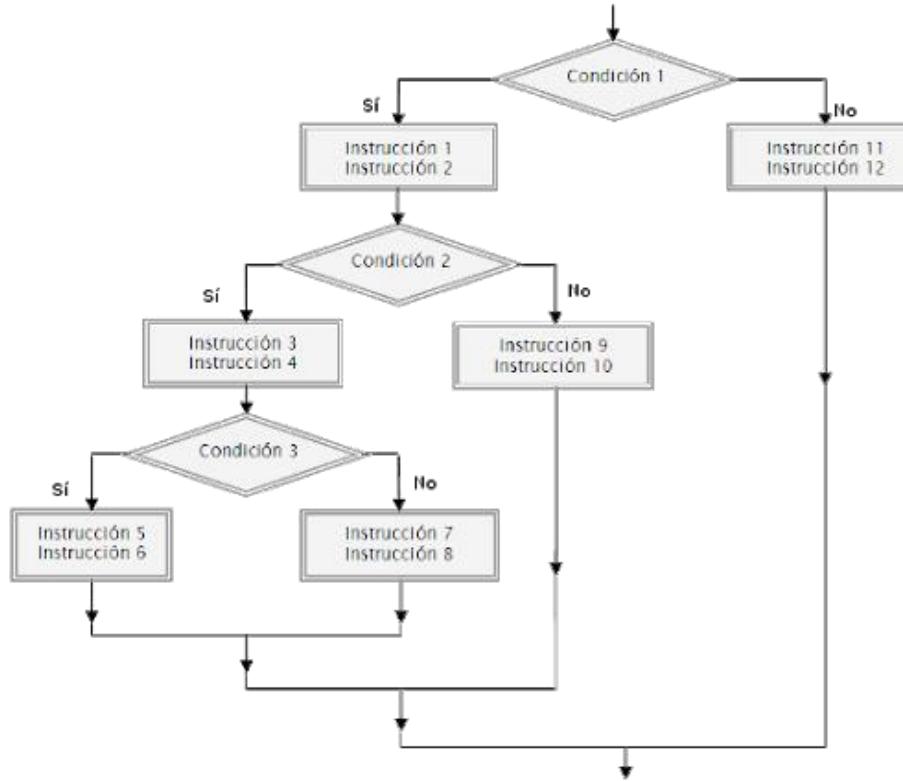
# Operadores relacionales

Permiten hacer comparaciones entre cantidades, constantes y variables.

Operador	Significado	Equivalente matemático
>	Mayor que	>
<	Menor que	<
>=	Mayor o igual que	$\geq$
<=	Menor o igual que	$\leq$
=	Igual a	=
<>	Diferente a	$\neq$

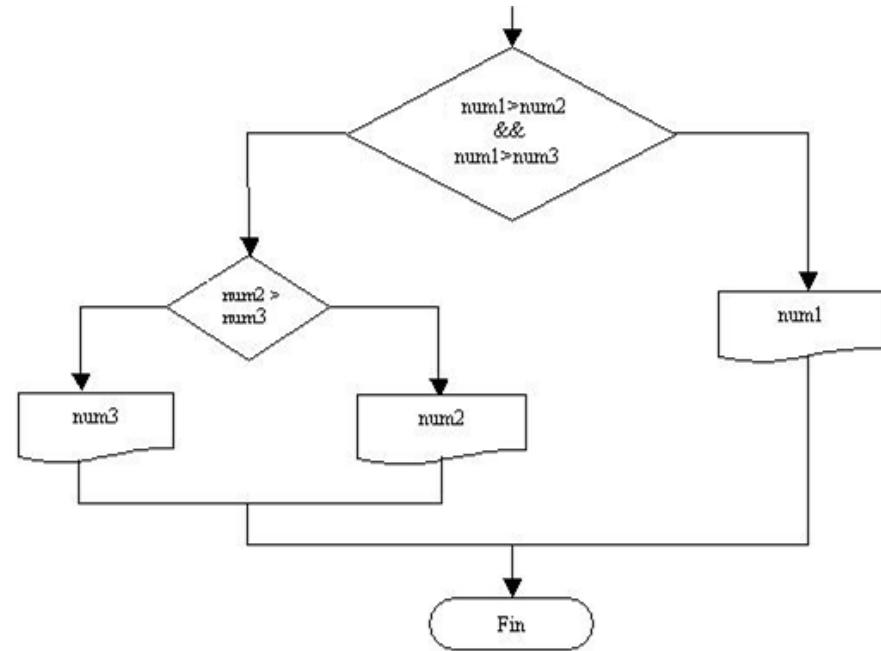
# Condicionales: Si, Si no

Ejemplos:



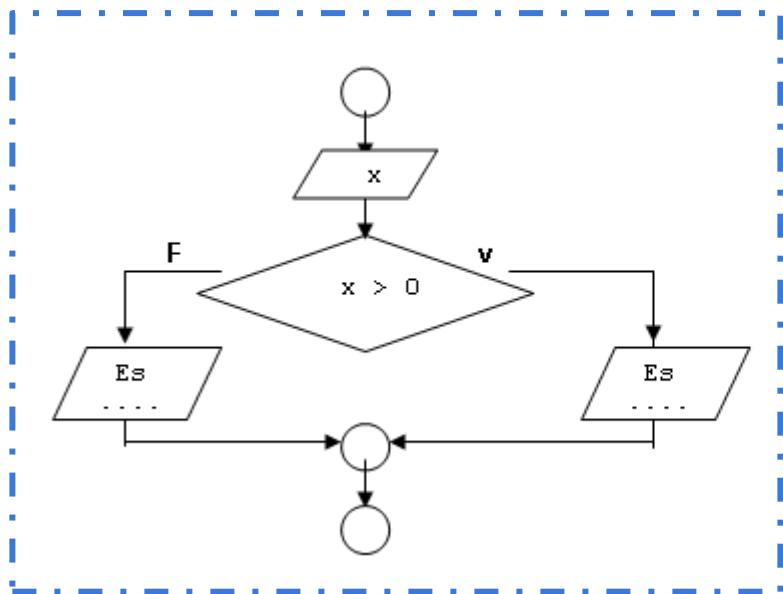
# Condicionales: Si, Si no

Los condicionales se usan para tomar decisiones al interior de un código



# Condicionales: Si, Si no

Los condicionales se usan para tomar decisiones al interior de un código

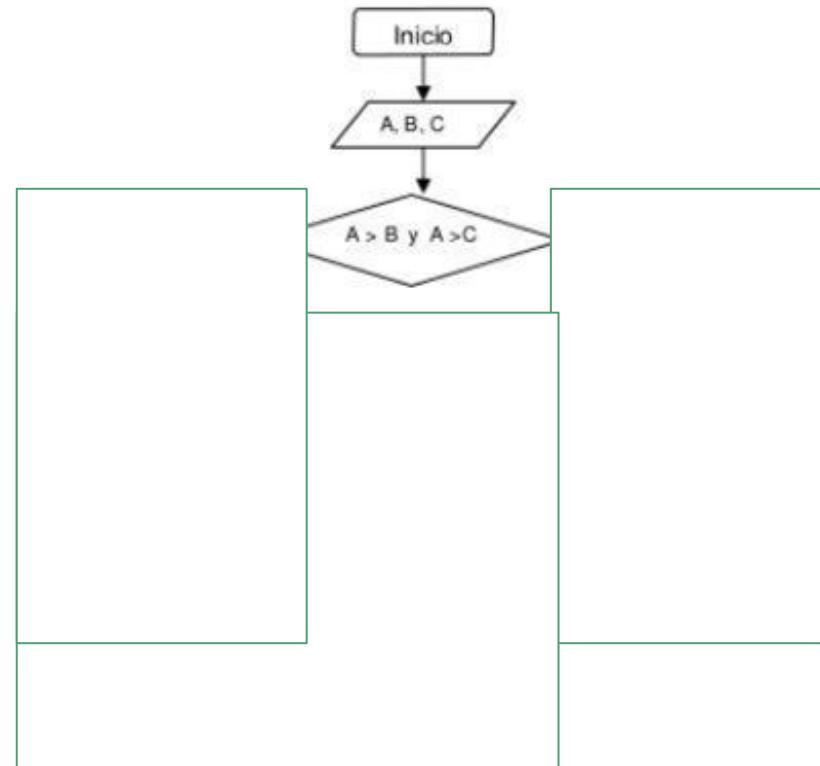


```
Inicio
  Entero x
  Leer x
  Si x > 0
    Imprimir "Es positivo"
  Sino
    Imprime "Es negativo"
  fin
```

# Condicionales: Si, Si no

Ejemplos:

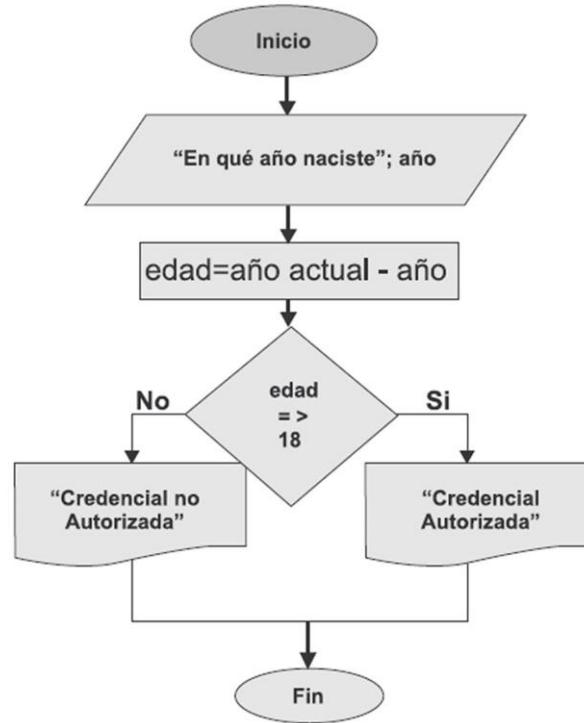
Calcular el número mayor  
entre 3 valores



# Condicionales: Si, Si no

Ejemplos:

¿Cómo saber si una persona es mayor de edad?

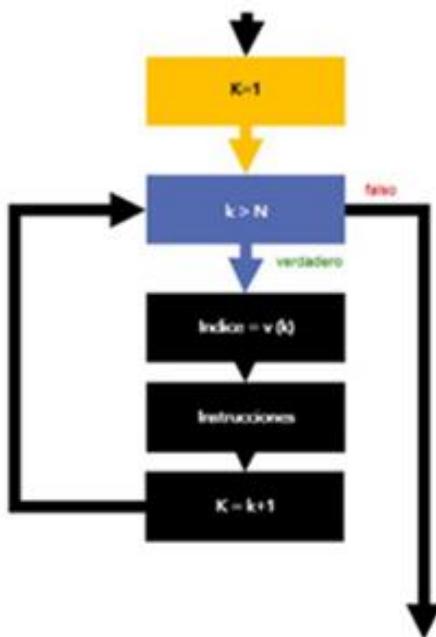


# Bucles: Para, Mientras



# Bucles: Para, Mientras

For



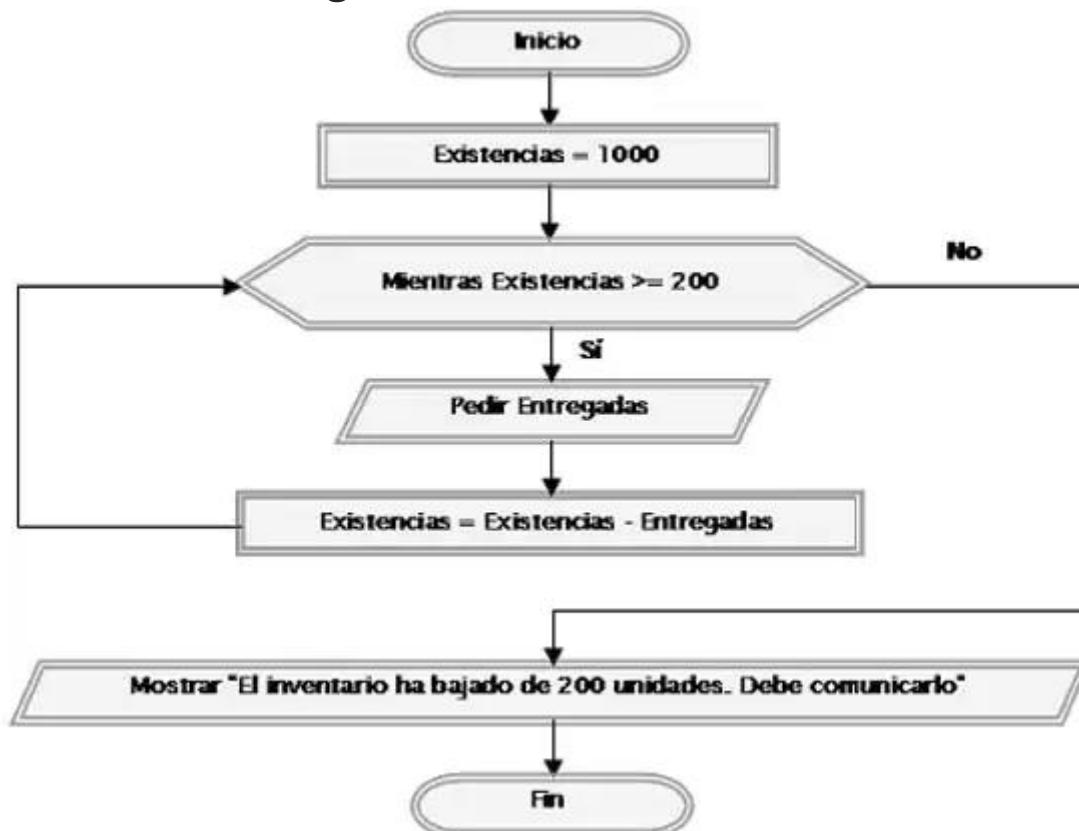
While



# Bucles y condicionales



# Diagrama de flujo



# Conclusiones

1. Los bucles son herramientas que permiten al programador realizar tareas de manera repetitiva por un número de veces que puede variar dependiendo de las necesidades.
2. Los condicionales le permiten al programador darle instrucciones al computador para que tome decisiones respecto a parámetros lógicos que dependerán del problema.