

CONQUER
BLOCKS

BASES DE LA INFORMÁTICA

CODIFICACION DE DATOS Y
CREACIÓN DE ALGORITMOS



SOFTWARE: CREACIÓN DE UN PROGRAMA

QUEREMOS ENSEÑAR A UN ORDENADOR COMO ENTRAR EN CASA

Sentencia Declarativa

Entremos en la casa

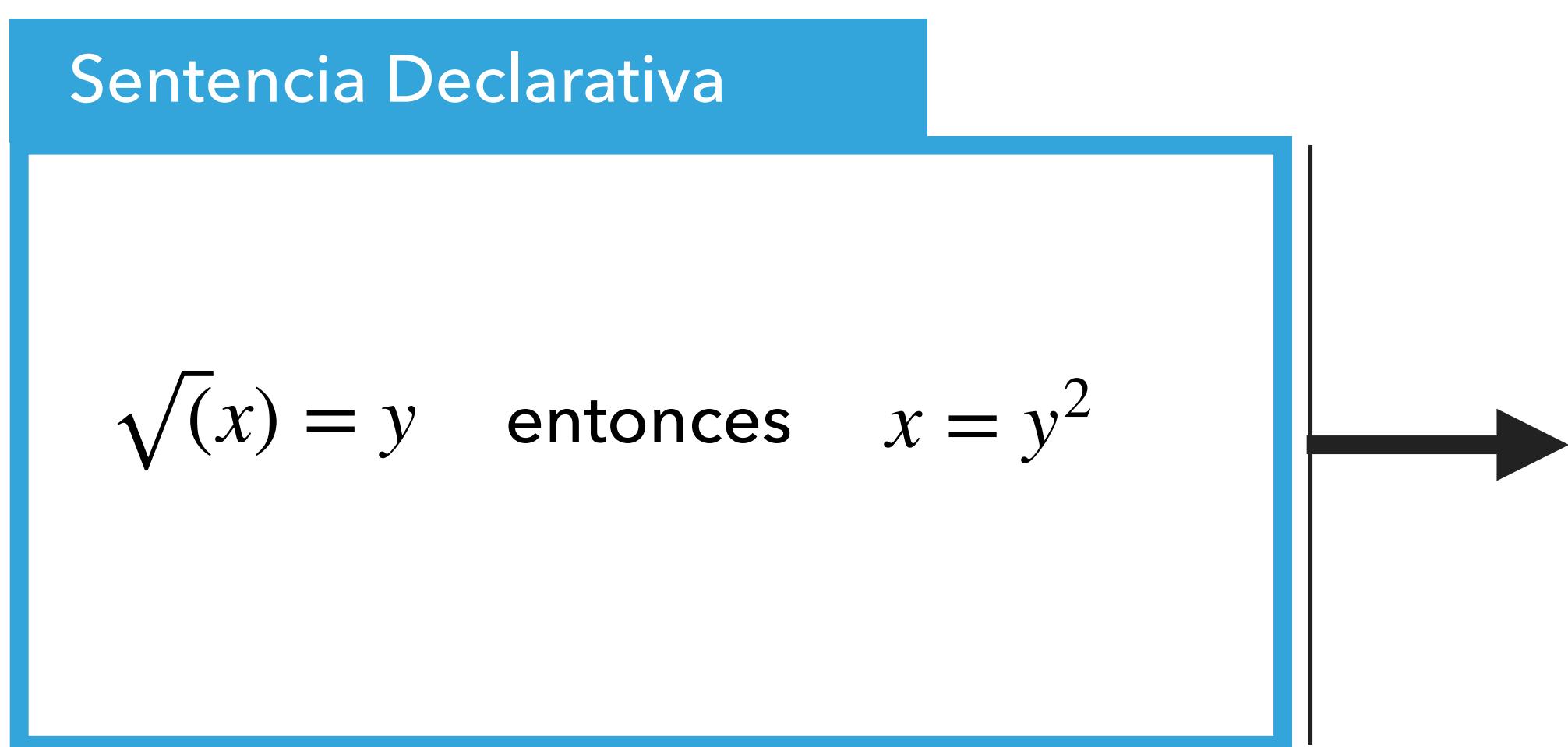


Sentencia Imperativa

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**'



SOFTWARE: CREACIÓN DE UN PROGRAMA



Sentencia Imperativa

1. Instrucción 1
 2. Instrucción 2
 3. Instrucción 3
 4. Instrucción 4
 5. Instrucción 5
- ...

Receta o secuencia de instrucciones específicas para que el ordenador haga el cálculo



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

Idioma artificial diseñado para expresar computaciones

Lenguaje de máquina



0110 1001 1010 1011

Lenguaje de bajo nivel

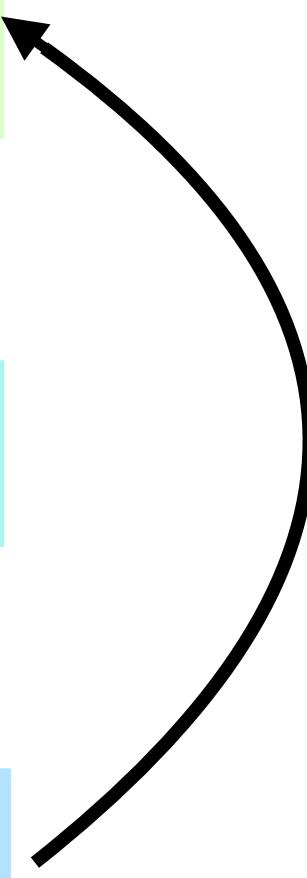


ADD M, N, P

Lenguaje de alto nivel

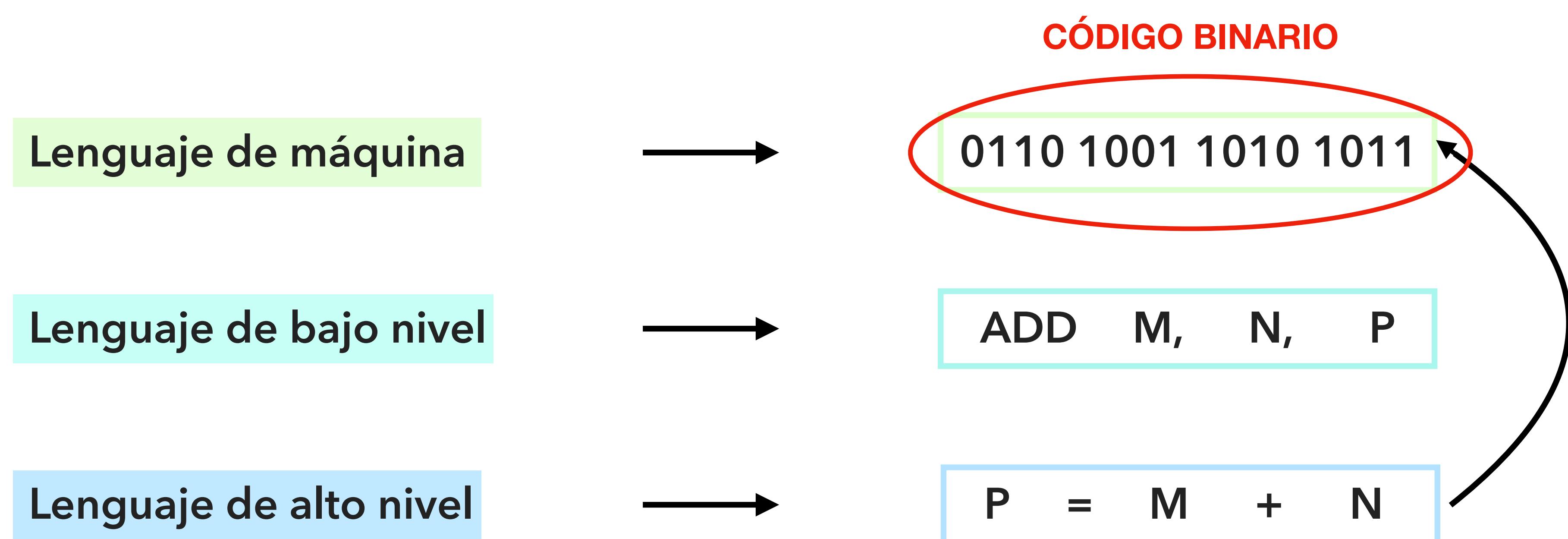


P = M + N



SOFTWARE: LENGUAJES DE PROGRAMACIÓN

Idioma artificial diseñado para expresar computaciones





CODIFICACIÓN DE DATOS

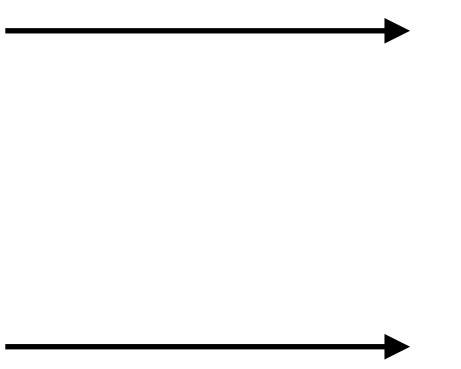
- Lenguaje hablado





CODIFICACIÓN DE DATOS

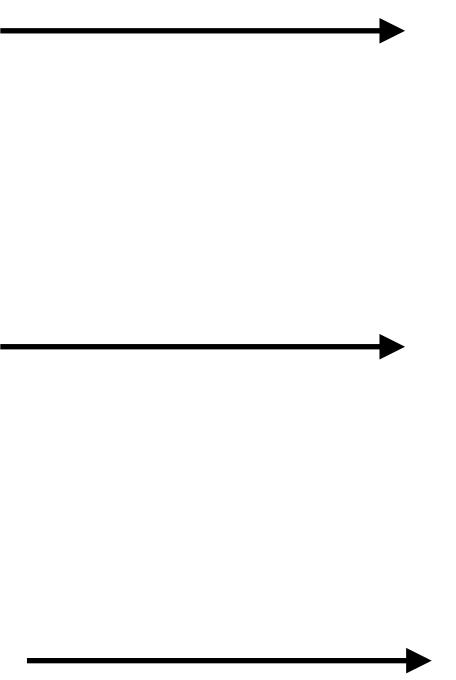
- Lenguaje hablado
- Gestos corporales





CODIFICACIÓN DE DATOS

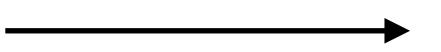
- Lenguaje hablado
- Gestos corporales
- Señales de tráfico





CODIFICACIÓN DE DATOS

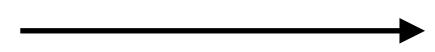
- Lenguaje hablado
- Gestos corporales
- Señales de tráfico
- Escritura





CODIFICACIÓN DE DATOS

- Lenguaje hablado
- Gestos corporales
- Señales de tráfico
- Escritura
- Morse



A =	• -	N =	- •
B =	- • • •	O =	---
C =	- • - •	P =	• - - •



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?

Tiempo



ENERO						
Lu	Ma	Mi	Ju	Vi	Sá	Do
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10



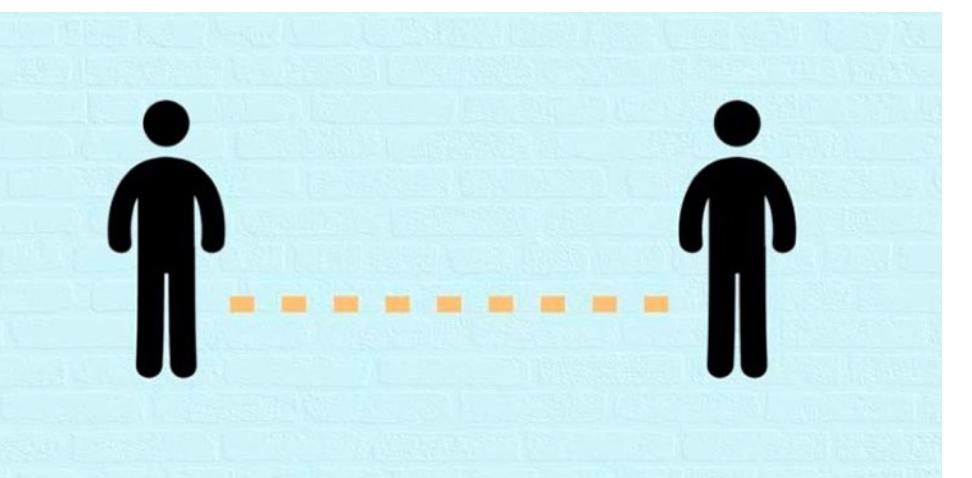
CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?

Tiempo



Posiciones



ENERO						
Lu	Ma	Mi	Ju	Vi	Sá	Do
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10



CODIFICACIÓN DE DATOS

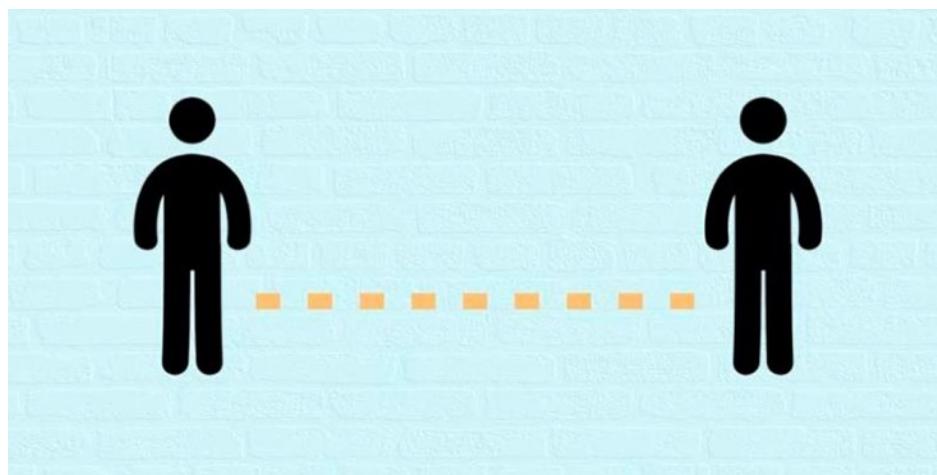
¿PODEMOS REDUCIR TODO A NÚMEROS?



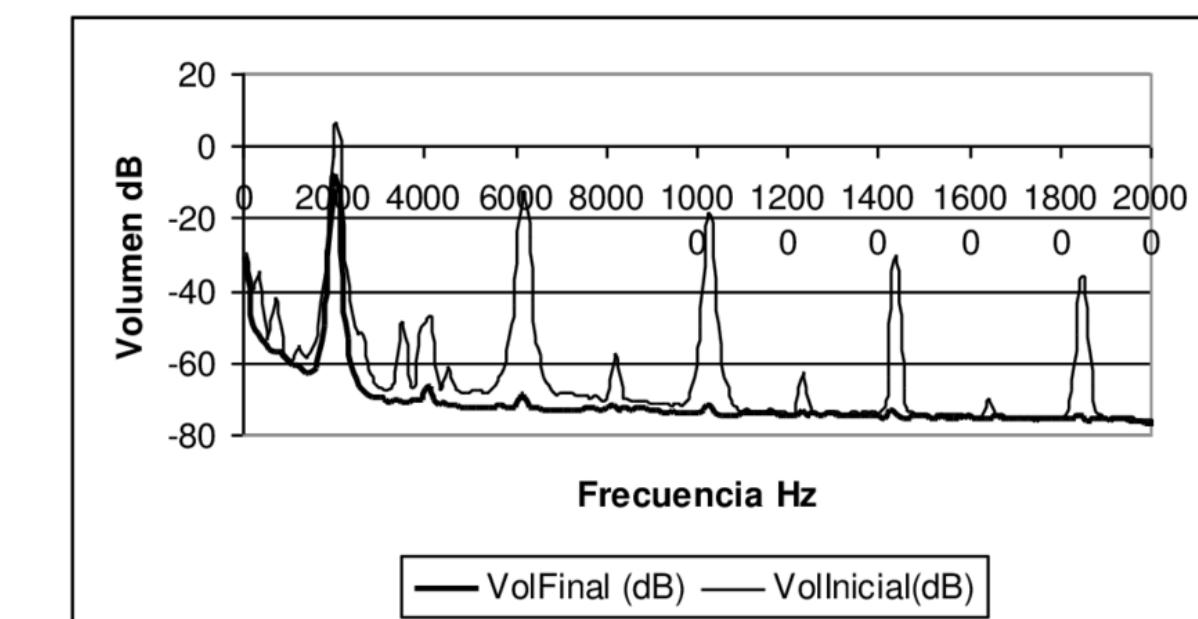
Tiempo



Posiciones



Sonido



ENERO						
Lu	Ma	Mi	Ju	Vi	Sá	Do
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?

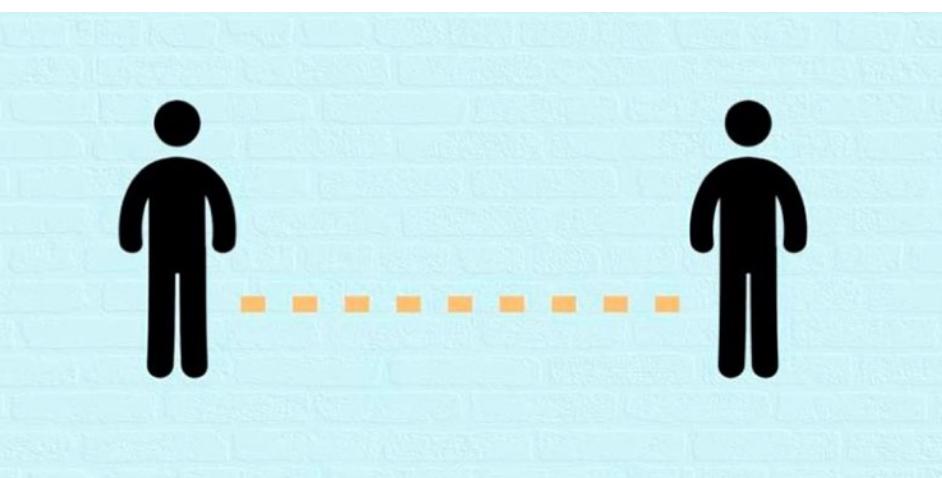


Tiempo

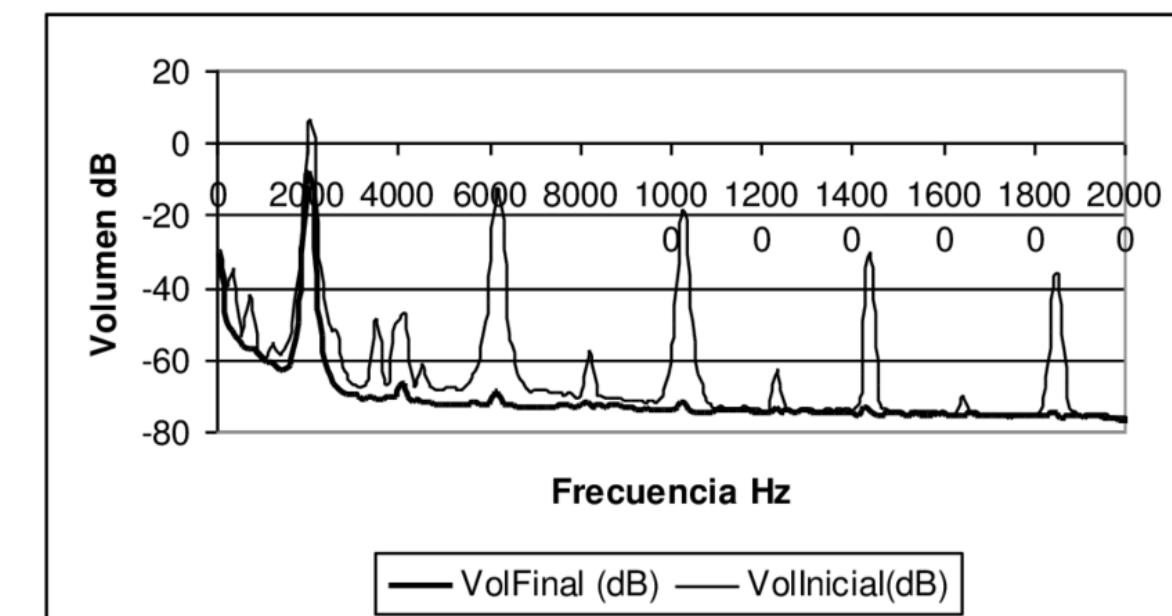


ENERO						
Lu	Ma	Mi	Ju	Vi	Sá	Do
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Posiciones



Sonido



Colores

1		RGB(0,0,0)
2		RGB(255,255,255)
3		RGB(0,255,0)
4		RGB(0,0,255)
5		RGB(0,204,255)
6		RGB(255,255,0)
7		RGB(255,0,255)
8		RGB(0,255,255)
9		RGB(128,0,0)
10		RGB(0,128,0)
11		RGB(0,0,128)
12		RGB(128,128,0)
13		RGB(128,0,128)
14		RGB(0,128,128)
15		RGB(192,192,192)
16		RGB(128,128,128)
17		RGB(153,153,255)
18		RGB(153,51,102)
19		RGB(255,255,204)
20		RGB(204,255,255)
21		RGB(102,0,102)
22		RGB(255,128,128)
23		RGB(0,102,204)
24		RGB(204,204,255)
25		RGB(0,0,128)
26		RGB(255,0,255)
27		RGB(255,255,0)
28		RGB(0,255,255)
29		RGB(128,0,128)
30		RGB(128,0,0)
31		RGB(0,128,128)
32		RGB(0,0,255)
33		RGB(0,204,255)
34		RGB(204,255,255)
35		RGB(255,255,0)
36		RGB(153,204,255)
37		RGB(128,128,255)
38		RGB(255,153,204)
39		RGB(204,153,255)
40		RGB(255,204,153)
41		RGB(51,102,255)
42		RGB(51,204,204)
43		RGB(153,204,0)
44		RGB(153,204,0)
45		RGB(255,153,0)
46		RGB(255,102,0)
47		RGB(102,102,153)
48		RGB(150,150,150)
49		RGB(0,51,102)
50		RGB(51,153,102)
51		RGB(0,51,0)
52		RGB(51,51,0)
53		RGB(153,51,0)
54		RGB(153,51,102)
55		RGB(51,51,153)
56		RGB(51,51,51)



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



100	10	1
		5



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



100	10	1
	1	5



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



100	10	1
2	1	5



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



100	10	1
2	1	5

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



215



100	10	1
2	1	5

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

SISTEMA DECIMAL



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?

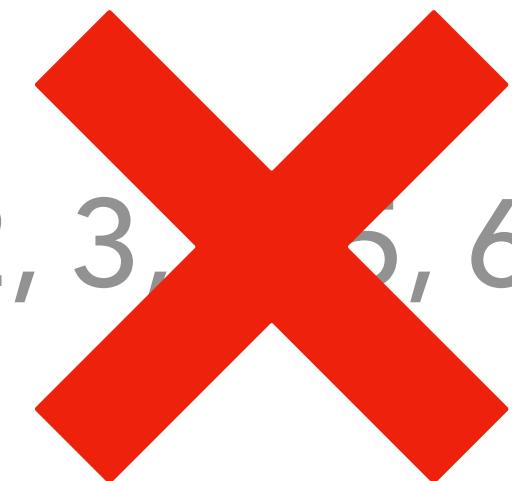


En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]





CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]

SISTEMA BINARIO



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]

SISTEMA BINARIO

SISTEMA DECIMAL

100	10	1
0	3	7



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]

SISTEMA BINARIO

SISTEMA DECIMAL

100	10	1
0	3	7

SISTEMA BINARIO

32	16	8	4	2	1



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]

SISTEMA BINARIO

SISTEMA DECIMAL

100	10	1
0	3	7

SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



En un circuito tenemos dos opciones:

O pasa corriente o no pasa corriente



[1 ,0]

SISTEMA BINARIO

SISTEMA DECIMAL

100	10	1
0	3	7

SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1

$$1 + 4 + 32 = 37$$



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



SISTEMA BINARIO					
32	16	8	4	2	1
1	0	0	1	0	1

$$1 + 4 + 32 = 37$$



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



CUALQUIER NÚMERO SE PUEDE EXPRESAR CON UNAS SERIE DE 0 Y 1



SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1

$$1 + 4 + 32 = 37$$



CODIFICACIÓN DE DATOS

¿PODEMOS REDUCIR TODO A NÚMEROS?



CUALQUIER NÚMERO SE PUEDE EXPRESAR CON UNAS SERIE DE 0 Y 1



SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1

$$1 + 4 + 32 = 37$$

PODEMOS REDUCIR TODO A CÓDIGO BINARIO





CÓDIGO BINARIO

PODEMOS REDUCIR TODO A CÓDIGO BINARIO



Tiempo

Posiciones

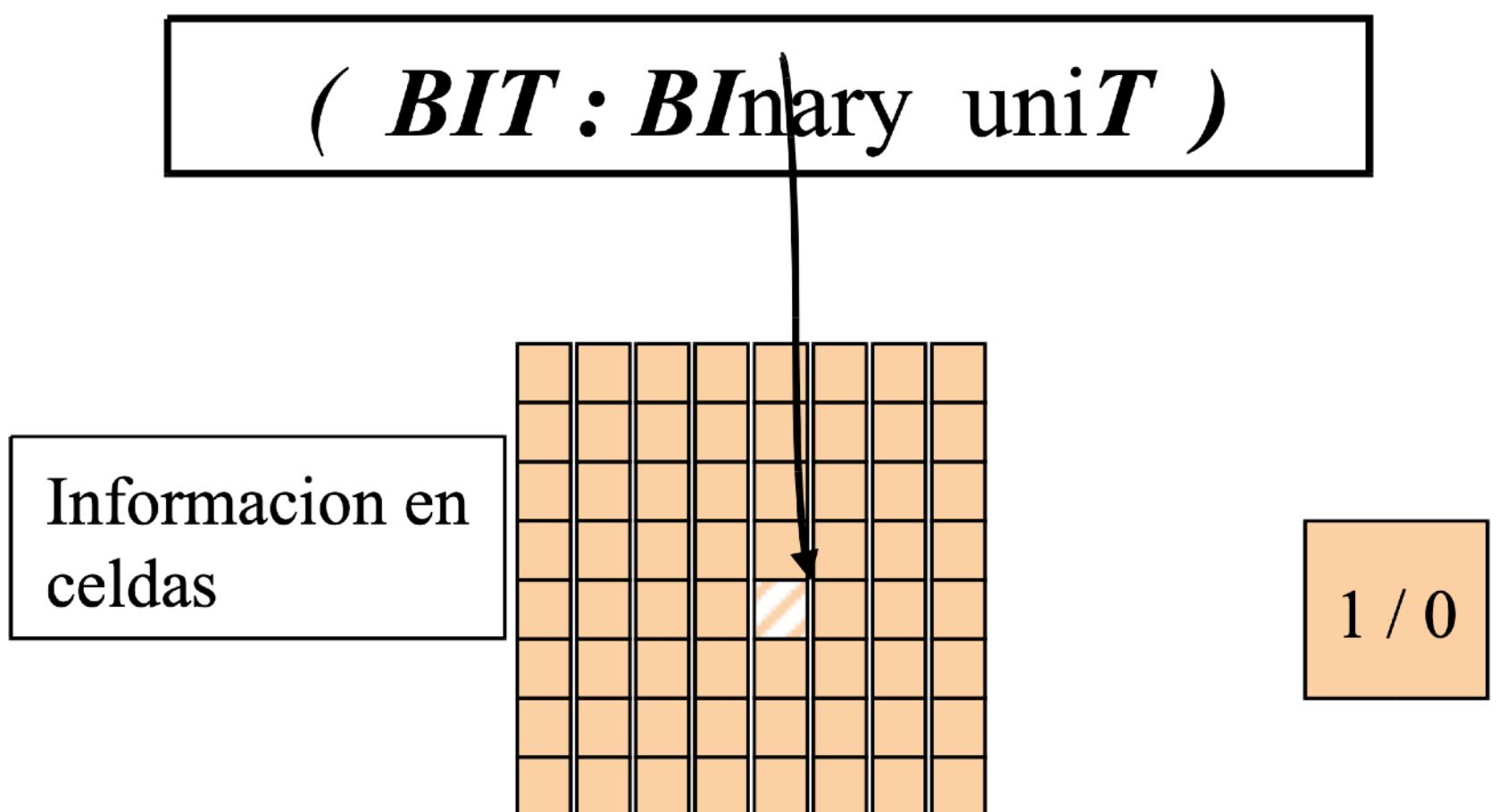
Colores

Sonidos

...

Podemos transmitir todo tipo de información usando este método de codificación

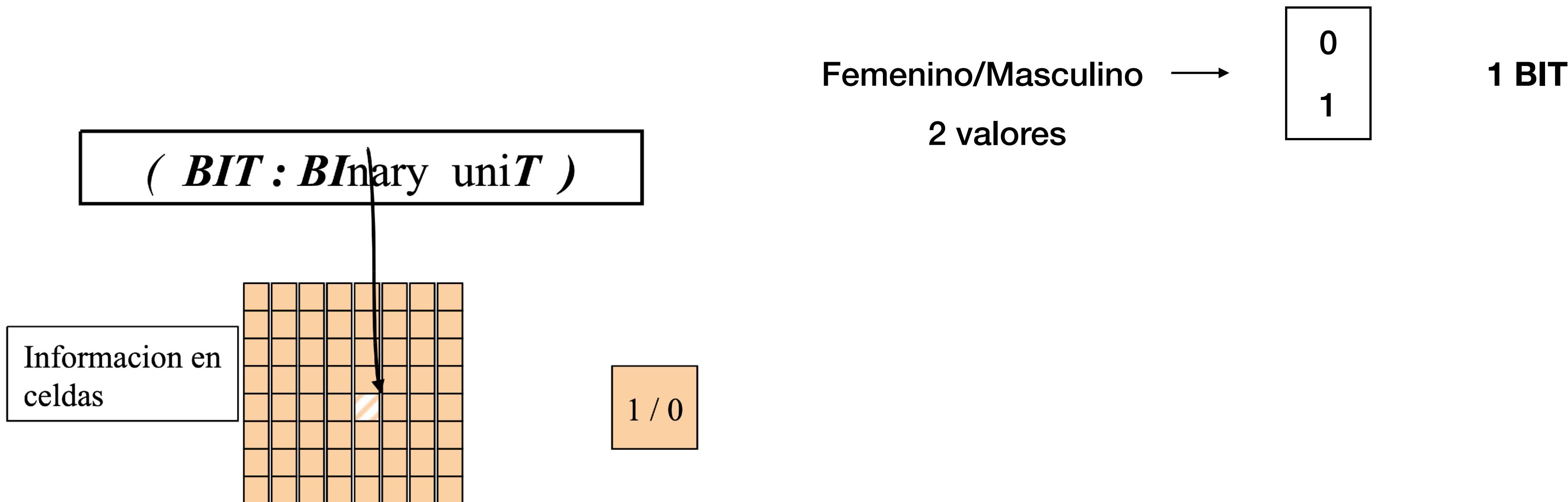
CÓDIGO BINARIO



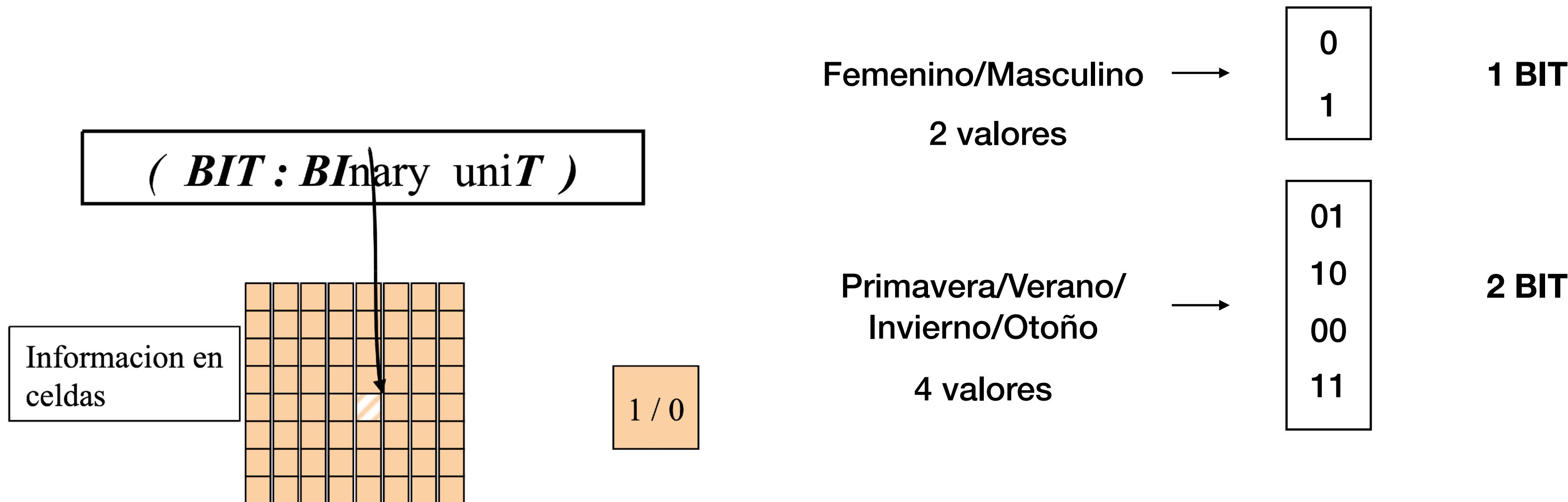
SISTEMA BINARIO

32	16	8	4	2	1
1	0	0	1	0	1

CÓDIGO BINARIO

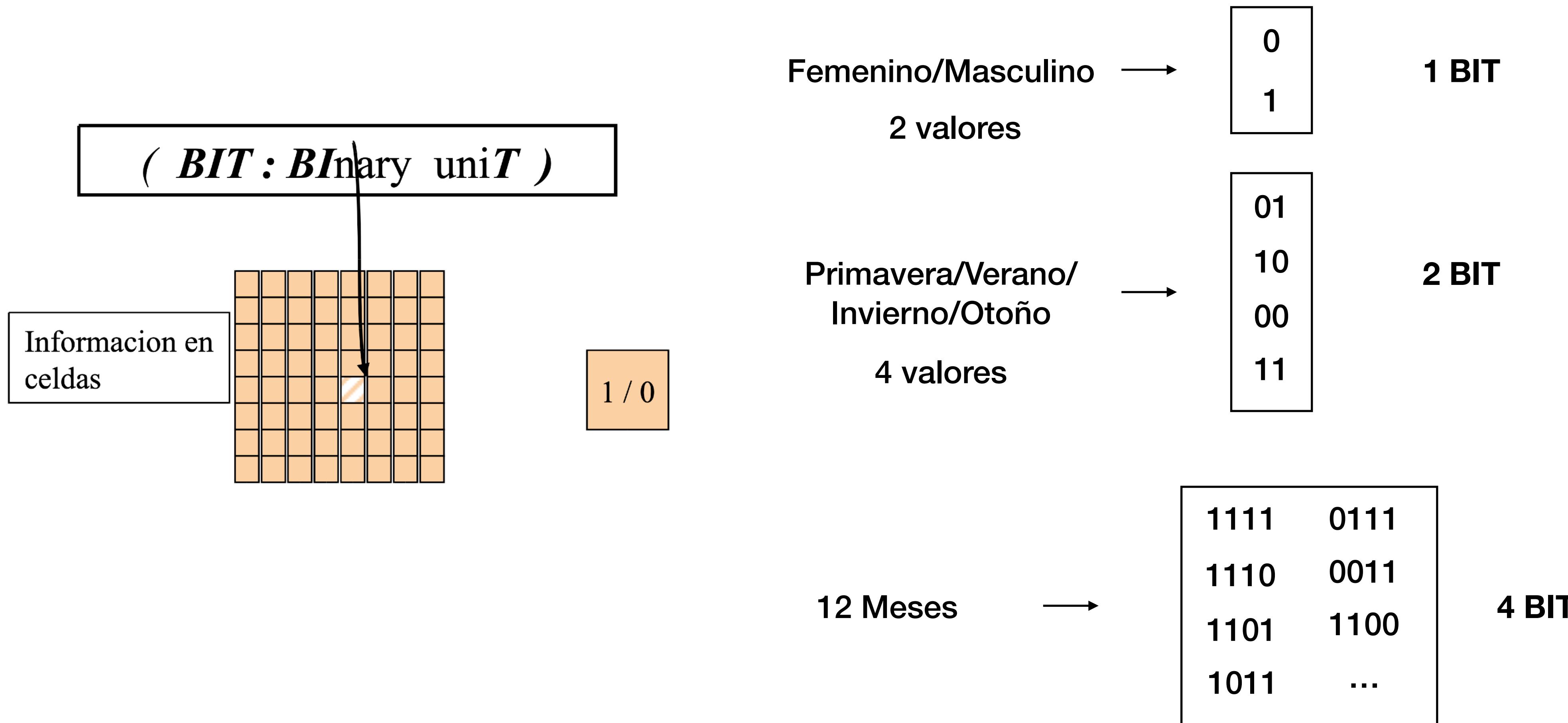


CÓDIGO BINARIO



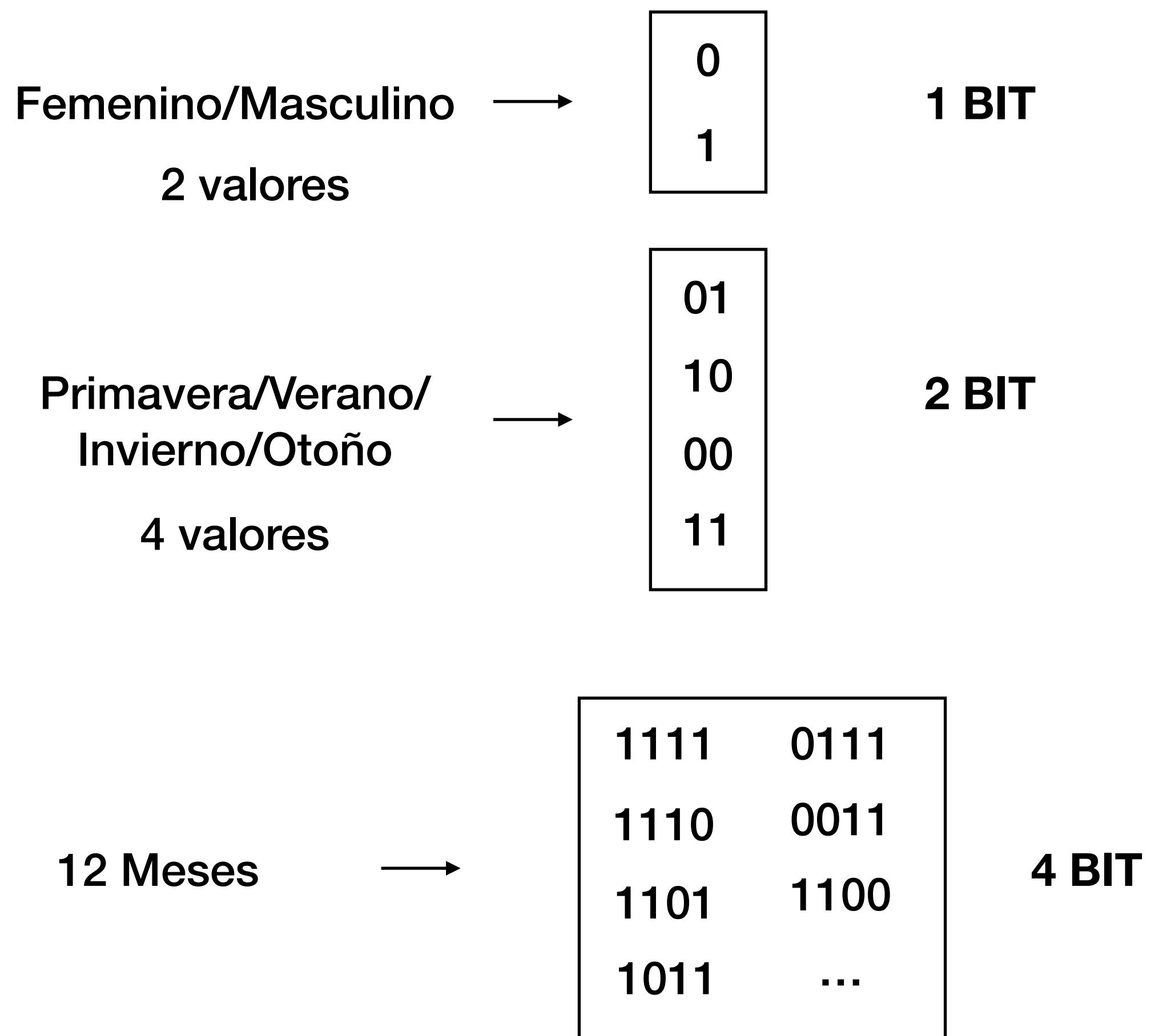


CÓDIGO BINARIO





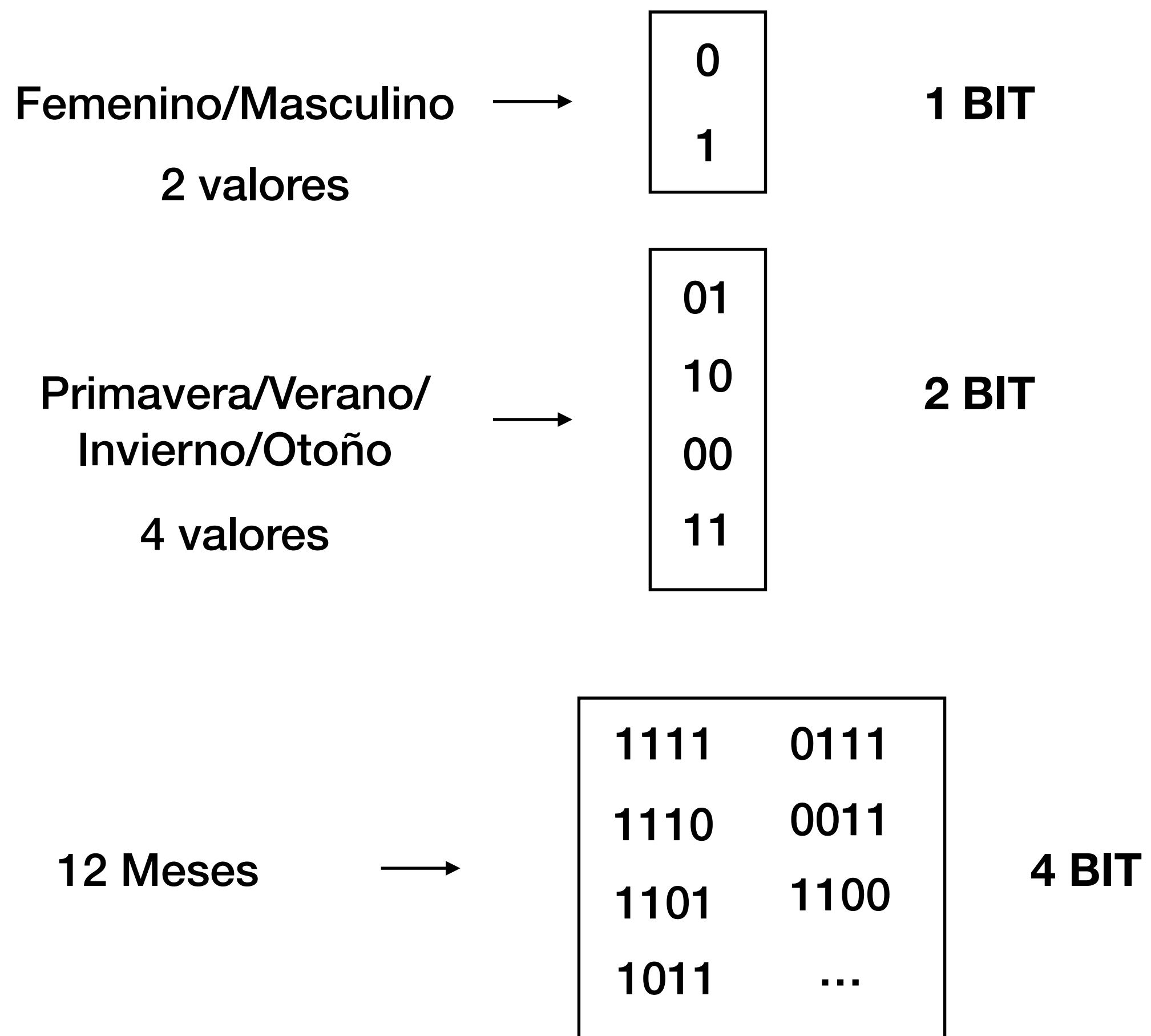
CÓDIGO BINARIO



n bits → 2^n datos



CÓDIGO BINARIO

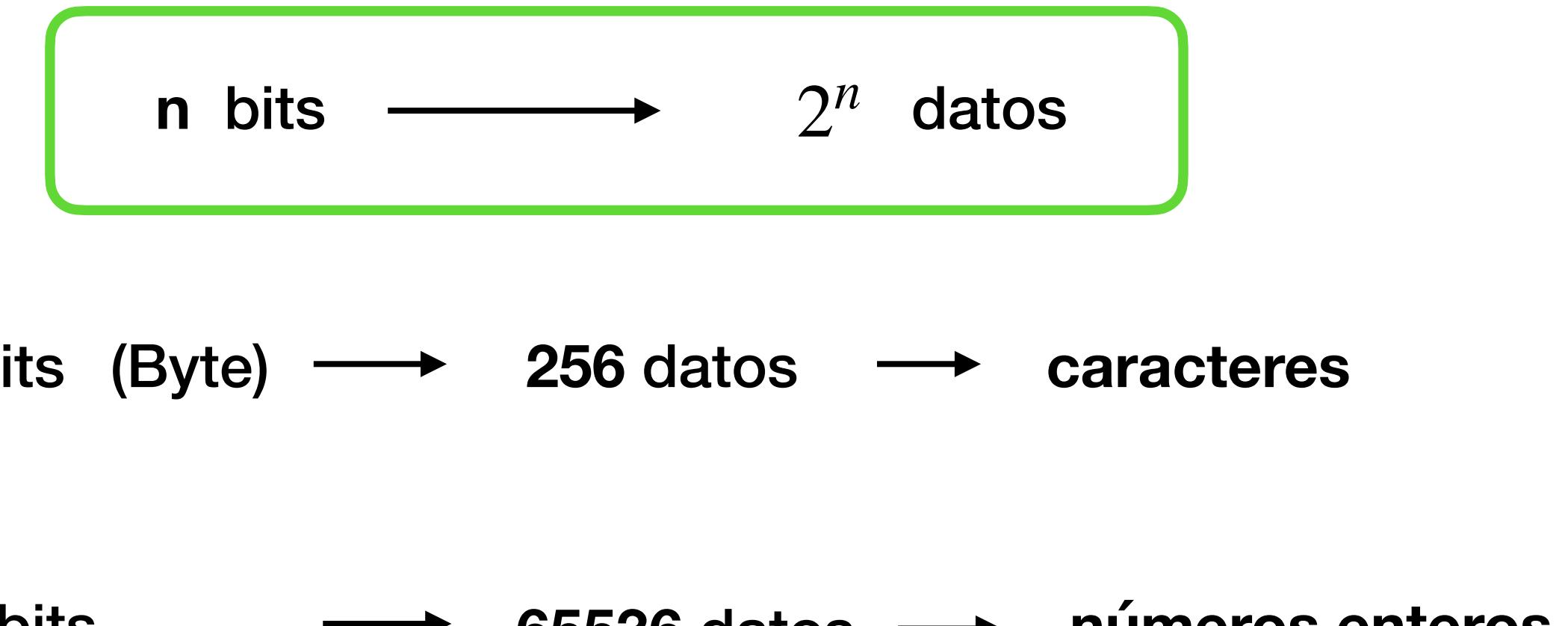
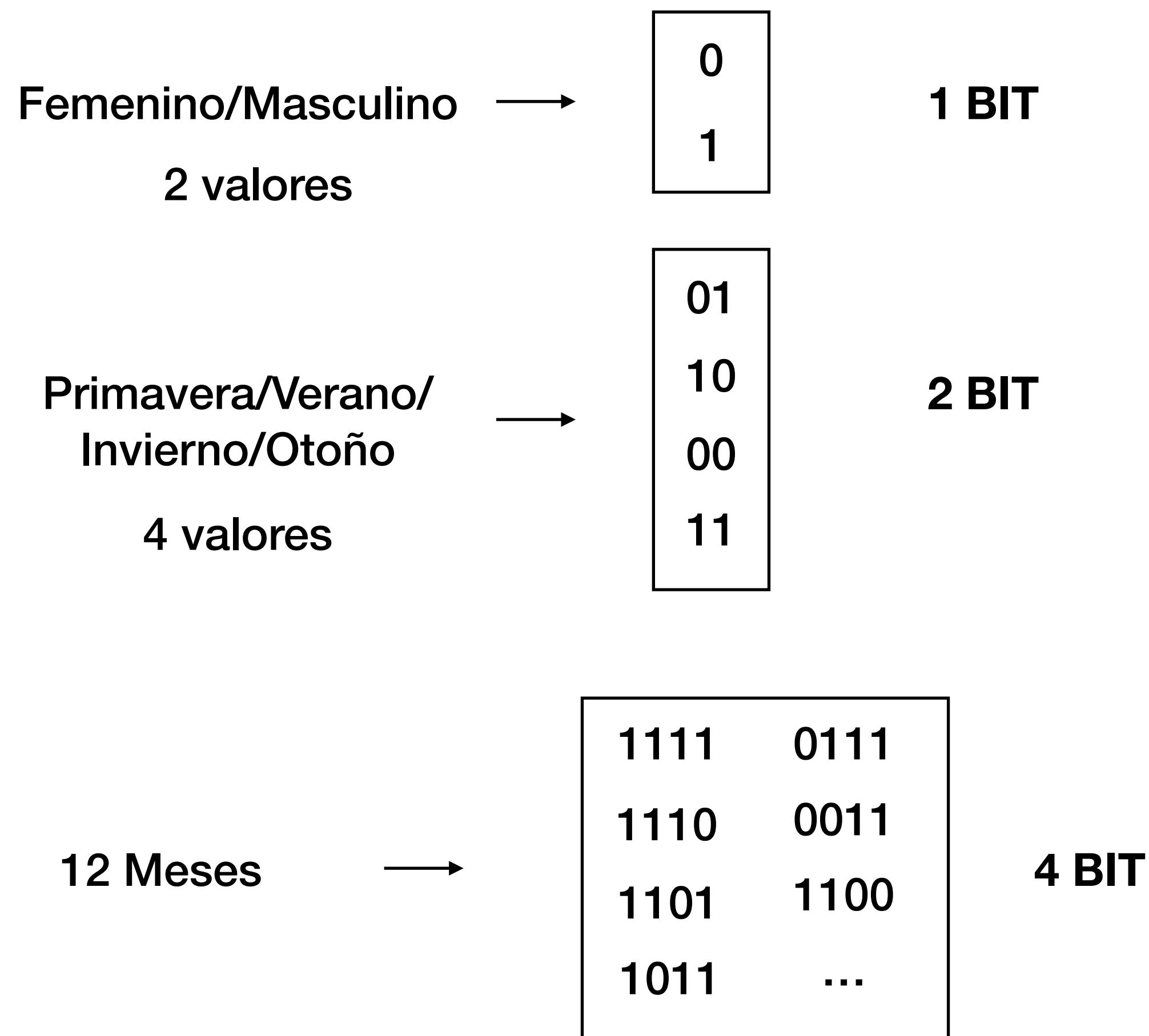


n bits → 2^n datos

8 bits (Byte) → 256 datos → caracteres

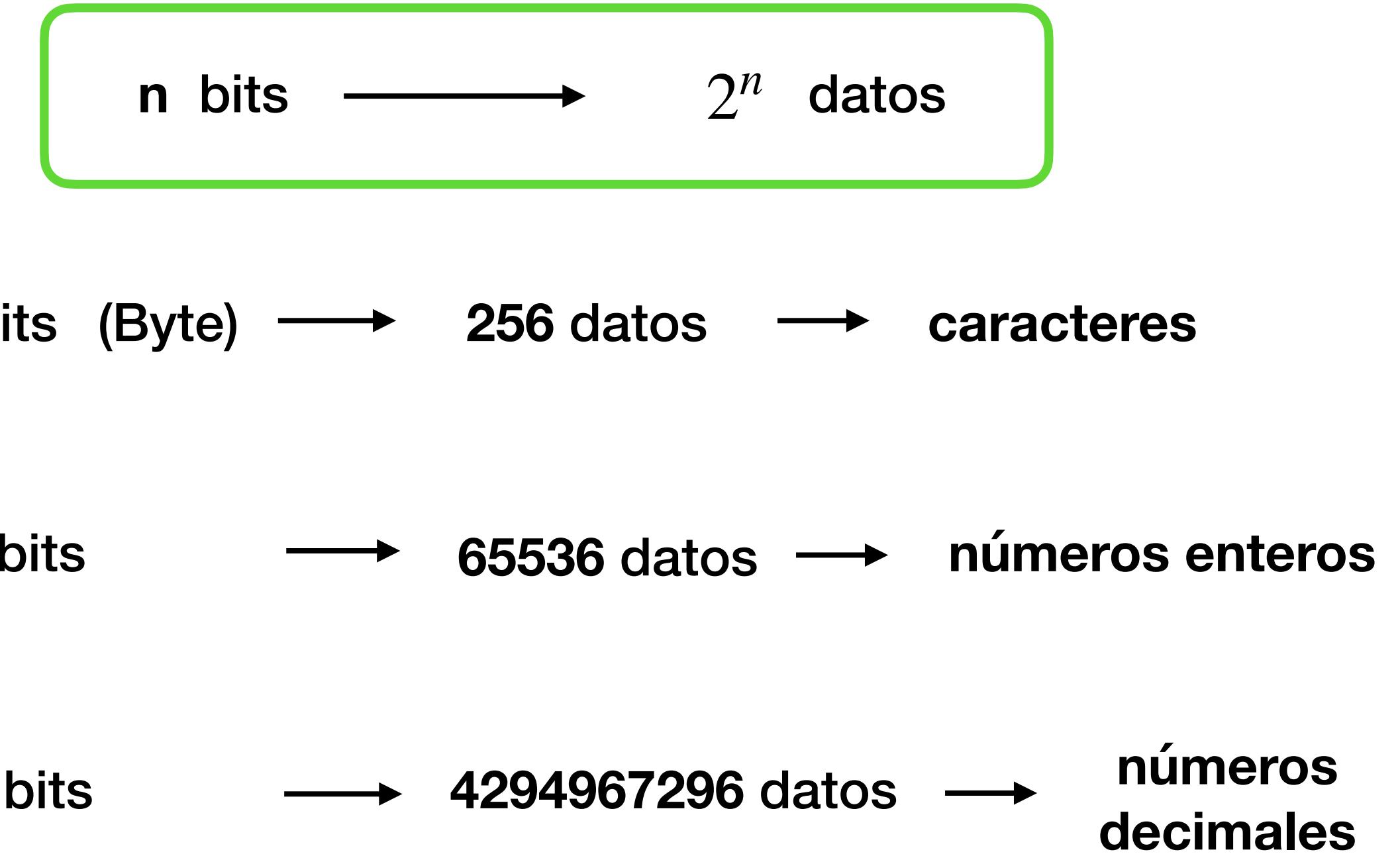
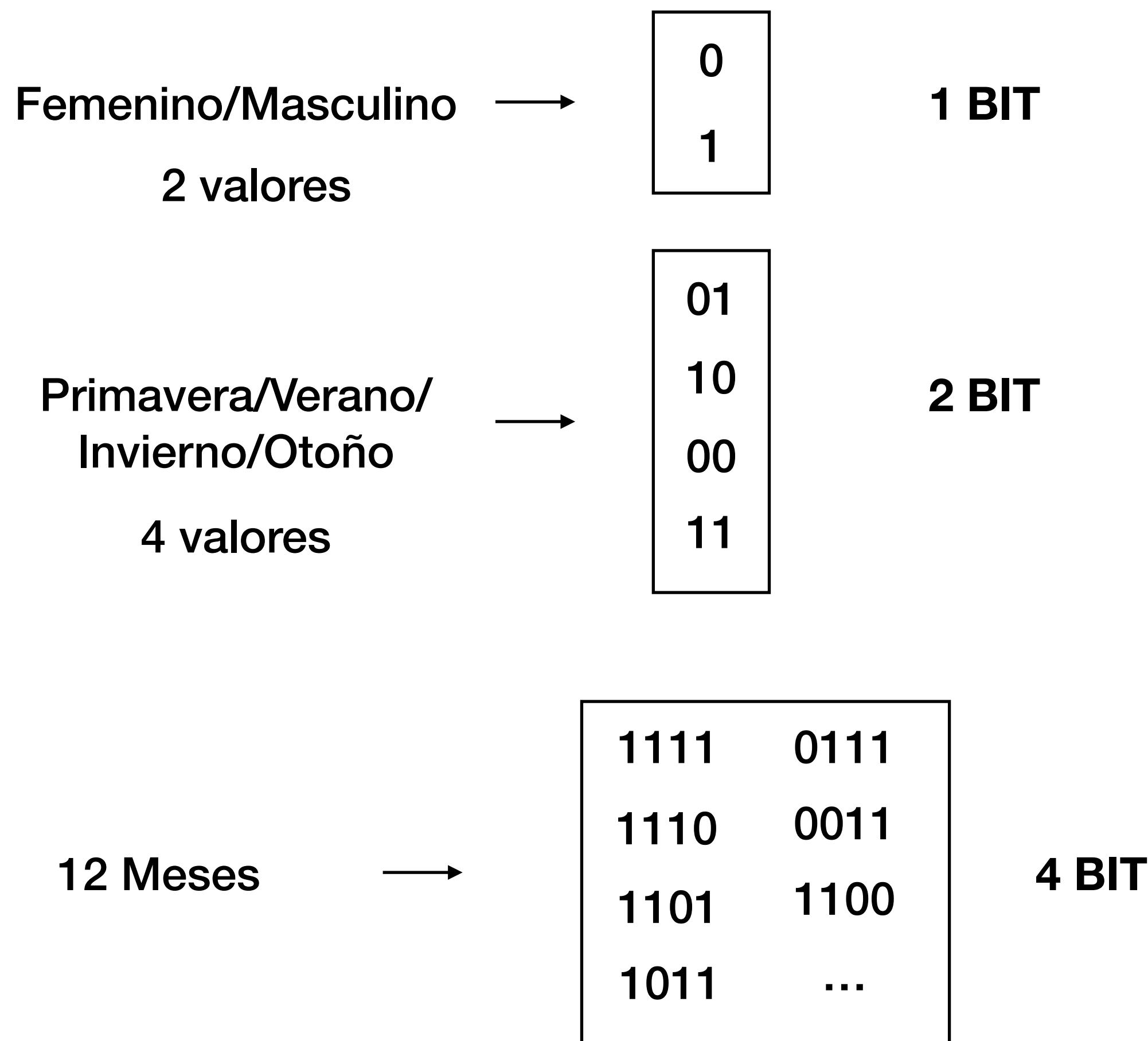


CÓDIGO BINARIO





CÓDIGO BINARIO





CÓDIGO BINARIO

Femenino/Masculino →

0
1

2 valores

1 BIT

Primavera/Verano/
Invierno/Otoño →

01
10
00
11

4 valores

2 BIT

12 Meses →

1111	0111
1110	0011
1101	1100
1011	...

4 BIT

n bits → 2^n datos

8 bits (Byte) → 256 datos → caracteres

16 bits → 65536 datos → números enteros

32 bits → 4294967296 datos → números decimales

número de bits/bytes → cantidad de información



CÓDIGO BINARIO



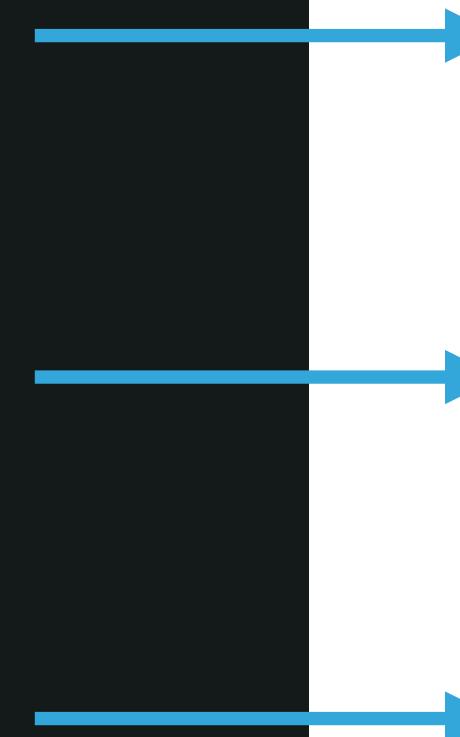
Capacidad. ¿Cuánto espacio necesitas?

128 GB²

256 GB²

512 GB²

¿No sabes qué capacidad elegir?
Hazte una idea de cuánto espacio necesitas.



SISTEMA BINARIO

64 32 16 8 4 2 1

$$64 \times 2 = 128$$

$$128 \times 2 = 256$$

$$256 \times 2 = 512$$



CÓDIGO BINARIO



Capacidad. ¿Cuánto espacio necesitas?

128 GB ²	42,04 €/mes o 1.009,00 €
256 GB ²	47,46 €/mes o 1.139,00 €
512 GB ²	58,29 €/mes o 1.399,00 €

¿No sabes qué capacidad elegir? +
Hazte una idea de cuánto espacio necesitas.



CÓDIGO BINARIO



Capacidad. ¿Cuánto espacio necesitas?

128 GB²	42,04 €/mes o 1.009,00 €
256 GB²	47,46 €/mes o 1.139,00 €
512 GB²	58,29 €/mes o 1.399,00 €

¿No sabes qué capacidad elegir? +

Hazte una idea de cuánto espacio necesitas.

1 KByte (KB) = 2^{10} (1024) byte
1 MegaByte (MB)= 2^{20} byte = 1024 KB
1 GigaByte (GB)= 2^{30} byte = 1024 MB
1 TeraByte (TB)= 2^{40} byte = 1024 GB



¿CÓMO PASAMOS A CÓDIGO BINARIO?

Lenguaje de máquina



0110 1001 1010 1011

Lenguaje de alto nivel



P = M + N





¿CÓMO PASAMOS A CÓDIGO BINARIO?

Lenguaje de máquina



0110 1001 1010 1011

Lenguaje de alto nivel



P = M + N

Compilador





¿CÓMO PASAMOS A CÓDIGO BINARIO?

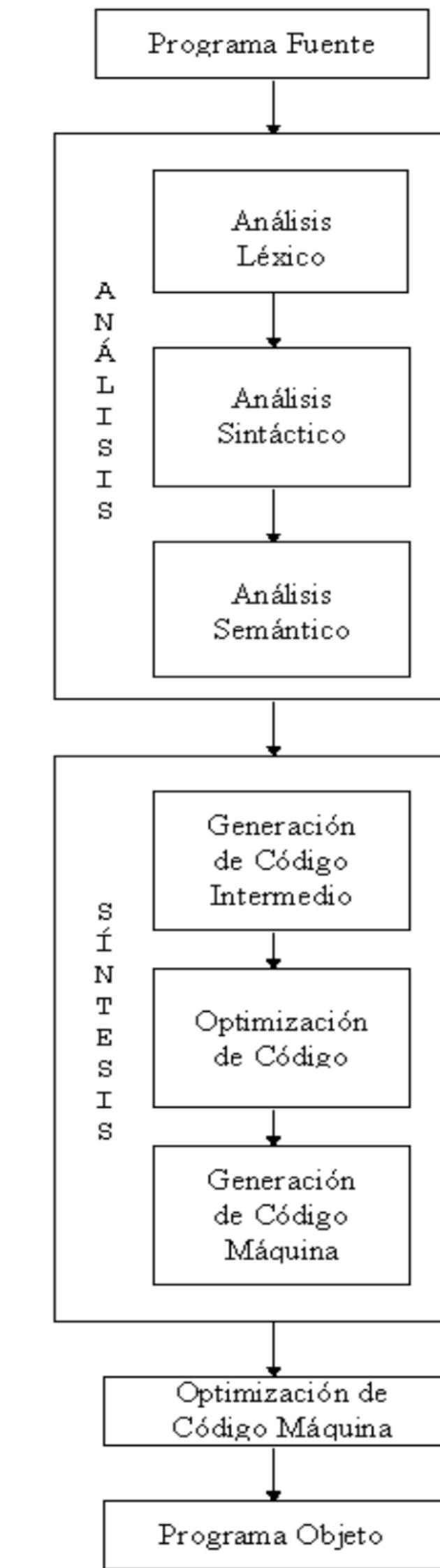
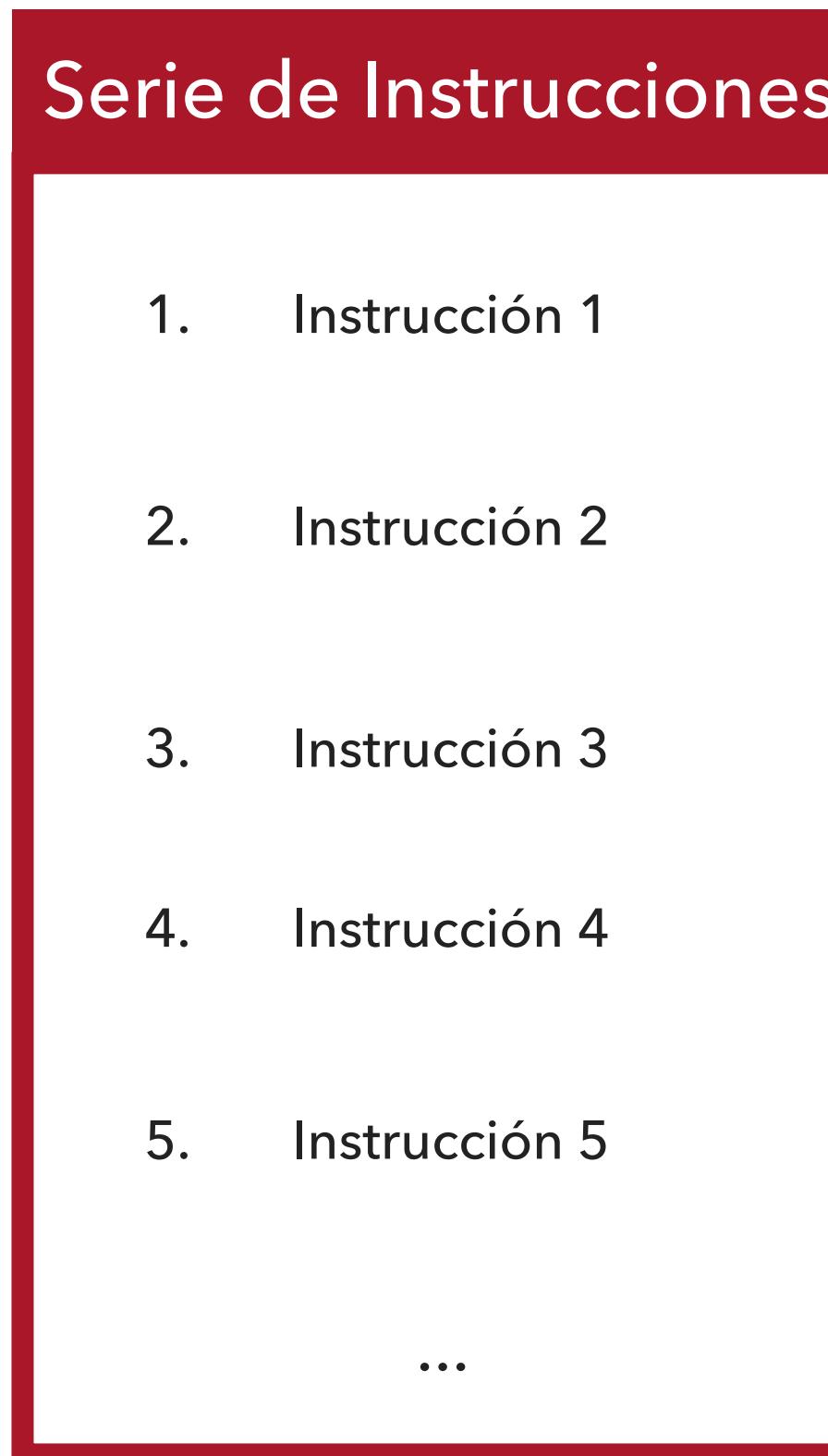
Compilador:

Es un traductor que convierte texto escrito en un lenguaje fuente de alto nivel en código máquina



¿CÓMO PASAMOS A CÓDIGO BINARIO?

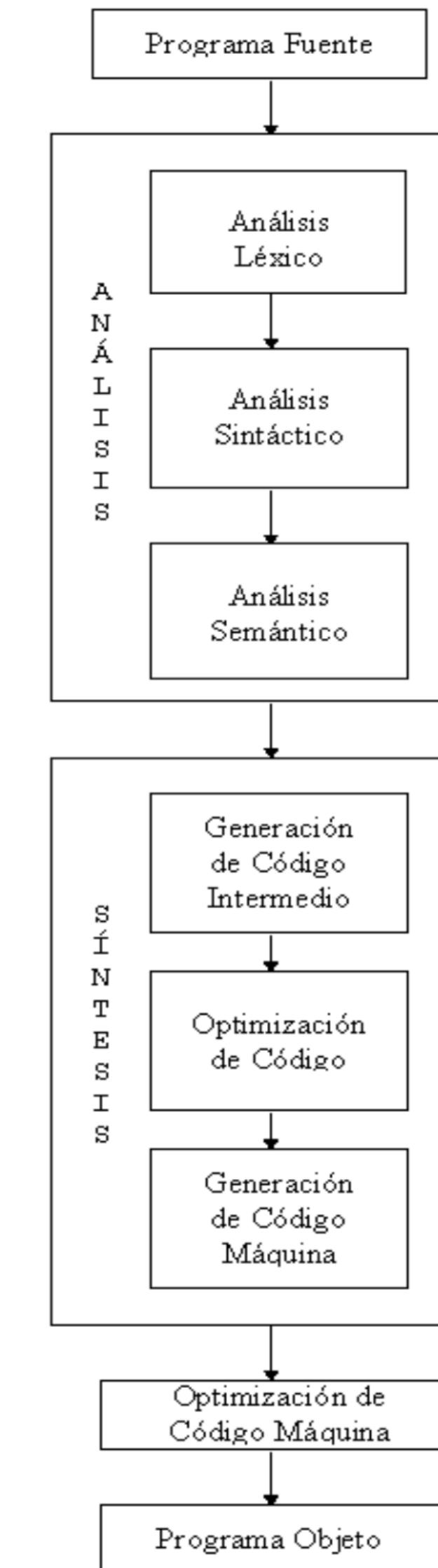
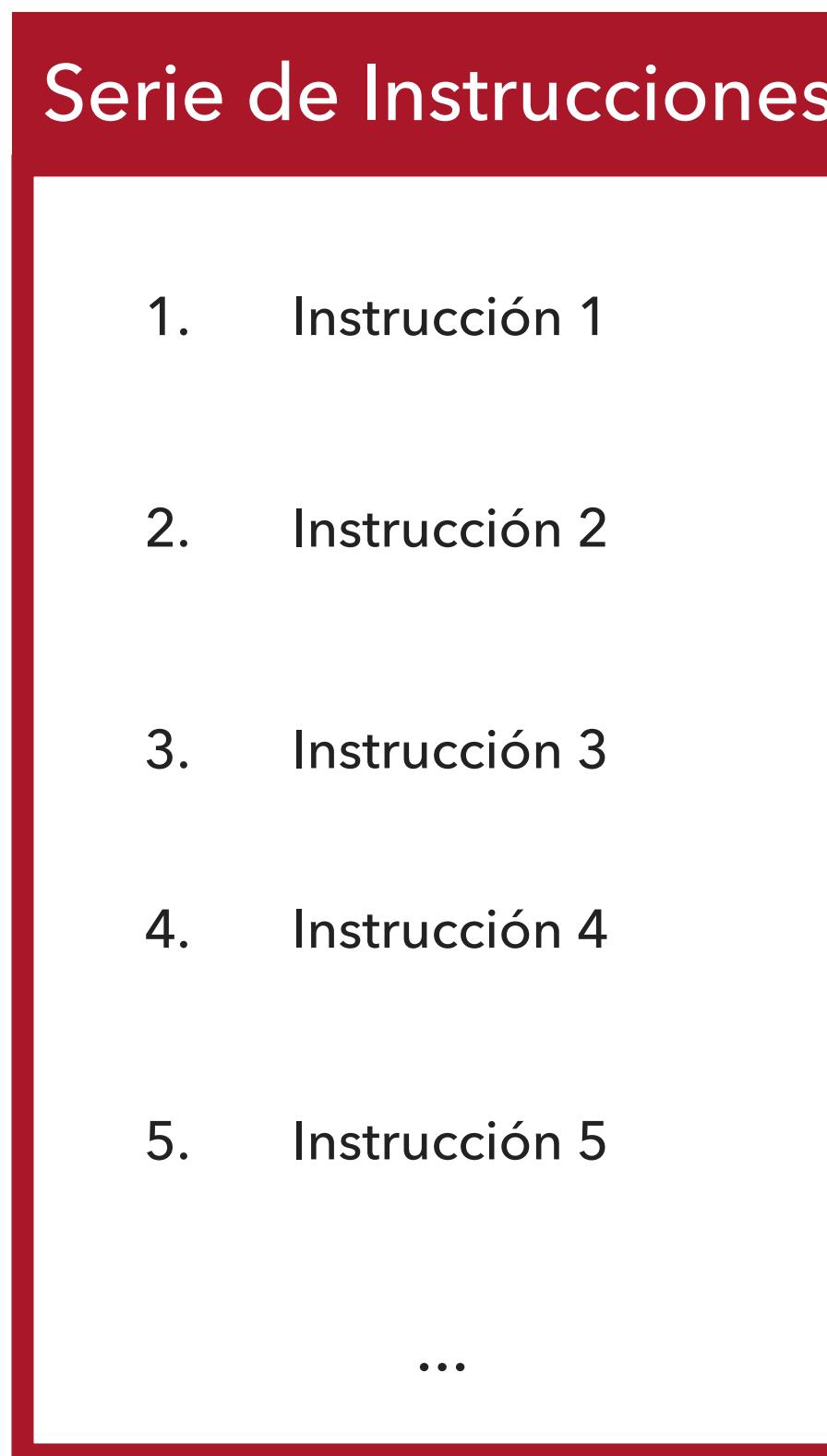
Esquema de un compilador:





¿CÓMO PASAMOS A CÓDIGO BINARIO?

Esquema de un compilador:



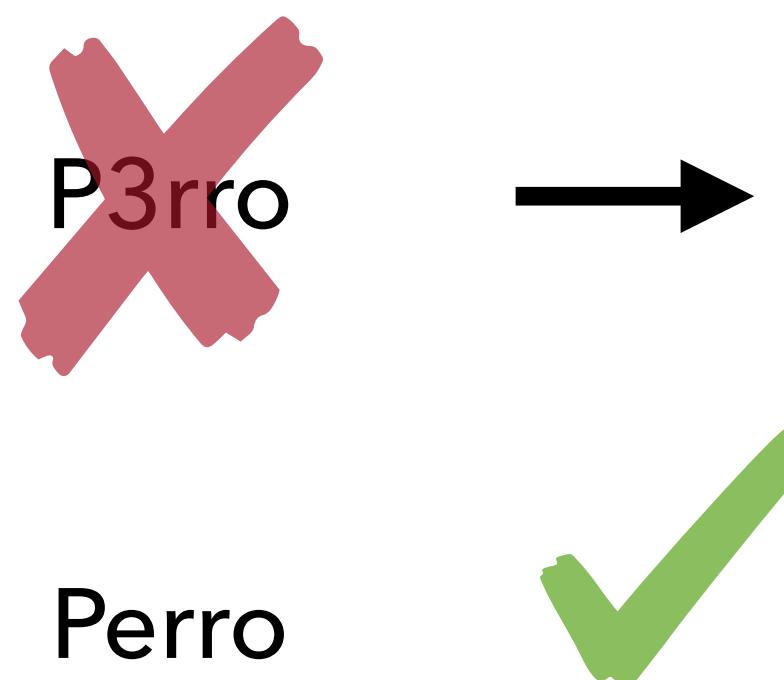
Analiza si las instrucciones están bien escritas y lo que significan



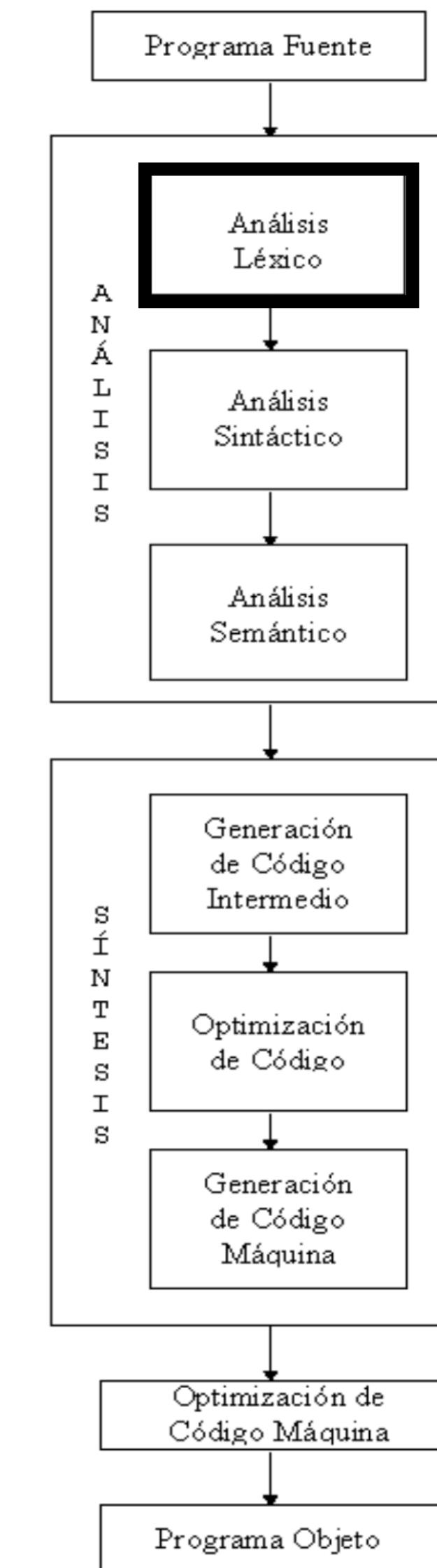
¿CÓMO PASAMOS A CÓDIGO BINARIO?

Esquema de un compilador:

Análisis léxico: ¿Las palabras que usamos existen en nuestro lenguaje?



**léxicamente
incorrecto**



Analiza si las instrucciones están bien escritas y lo que significan



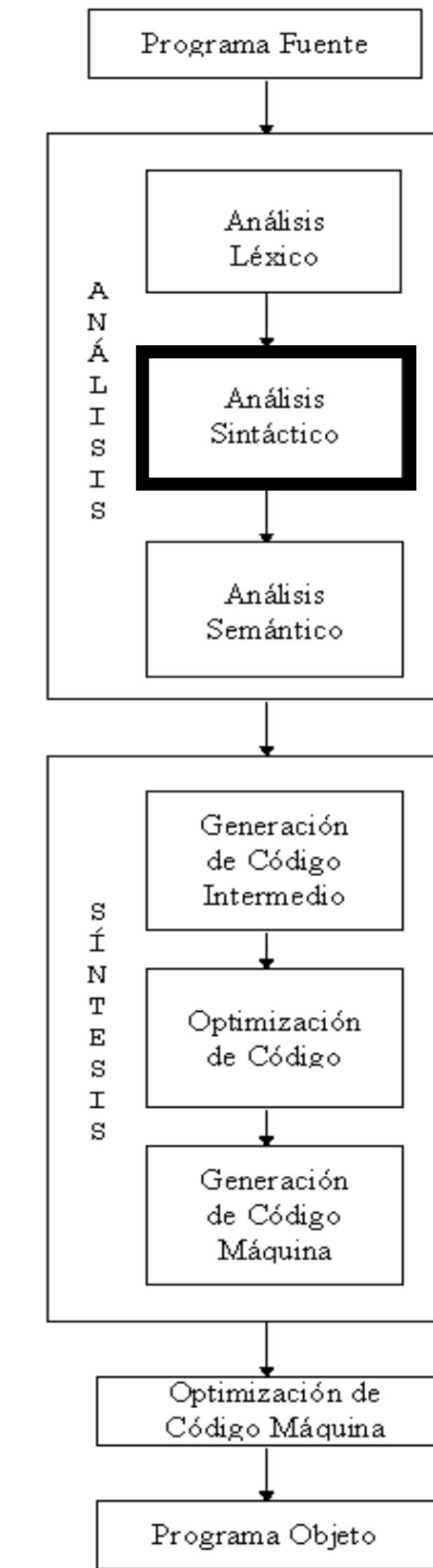
¿CÓMO PASAMOS A CÓDIGO BINARIO?

Esquema de un compilador:

Los lenguajes tienen normas de uso. A el conjunto de normas se les denomina **sintaxis**

Perro Juan jugar  → **sintácticamente incorrecto**

El perro juega con Juan 



Analiza si las instrucciones están bien escritas y lo que significan



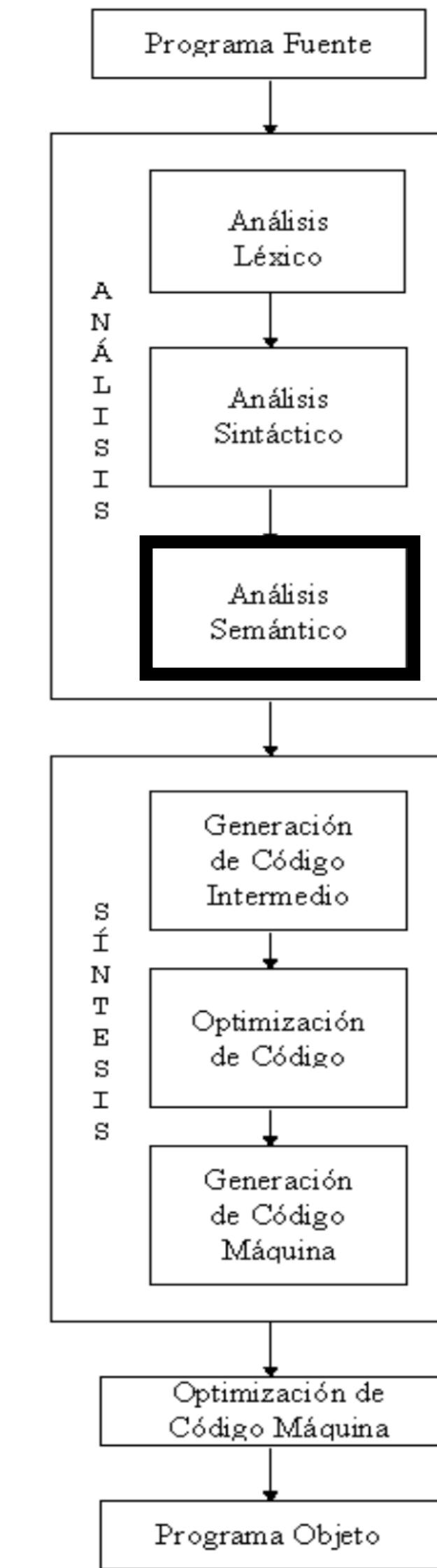
¿CÓMO PASAMOS A CÓDIGO BINARIO?

Esquema de un compilador:

Lo que decimos con el lenguaje tiene significado. A esto le denominamos **semántica**

La mesa se llama Juan → **sintácticamente correcto**
semánticamente incorrecto

Estos errores rara vez los encuentra el compilador!

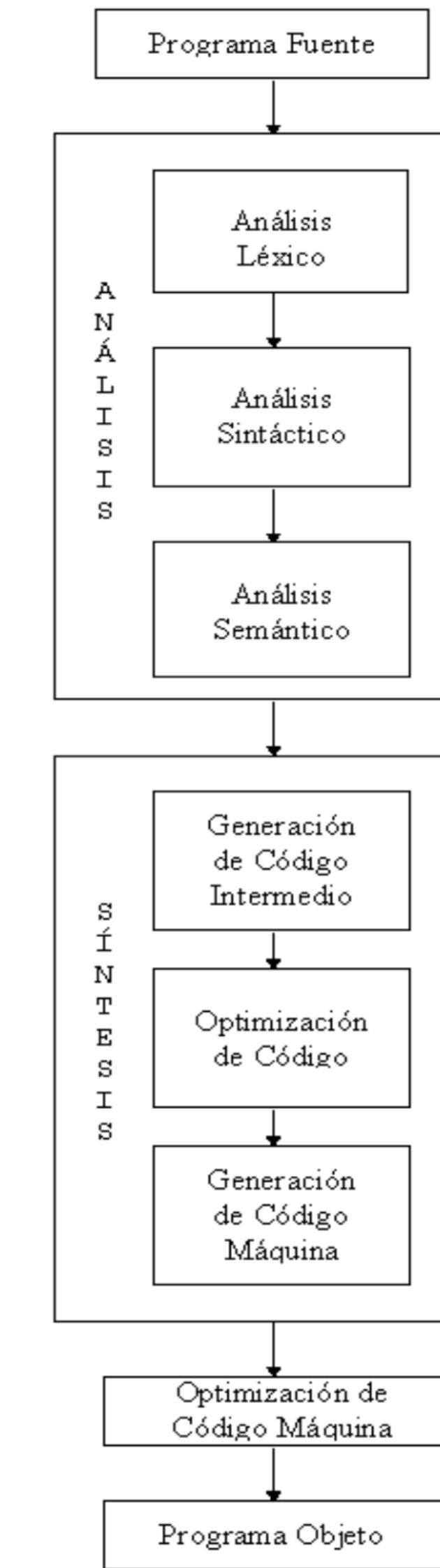
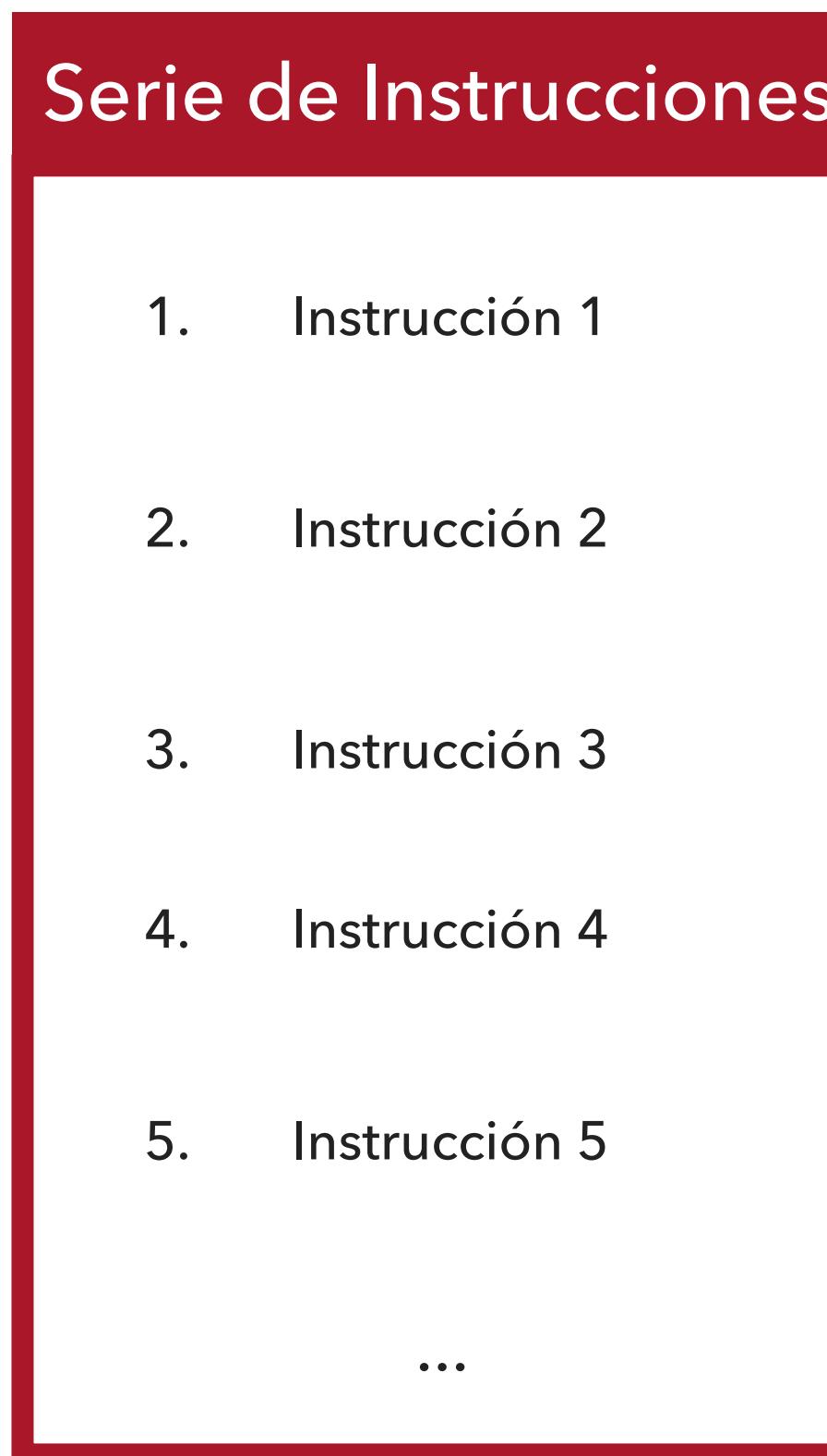


Analiza si las instrucciones están bien escritas y lo que significan



¿CÓMO PASAMOS A CÓDIGO BINARIO?

Esquema de un compilador:



Analiza si las instrucciones están bien escritas y lo que significan

↓
Genera la traducción del código



¿CÓMO PASAMOS A CÓDIGO BINARIO?

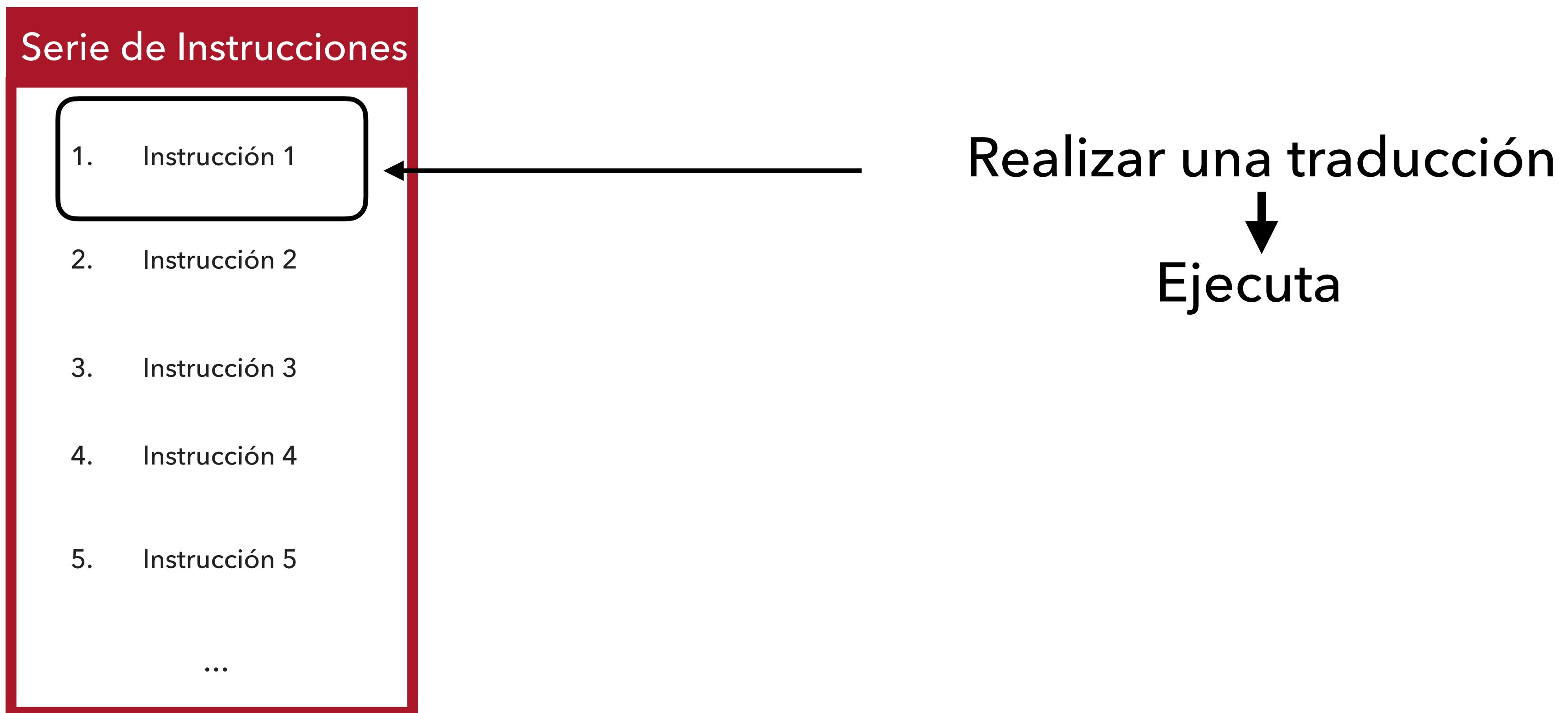
Intérprete:

Traductor que realiza la operación de compilación
paso a paso



¿CÓMO PASAMOS A CÓDIGO BINARIO?

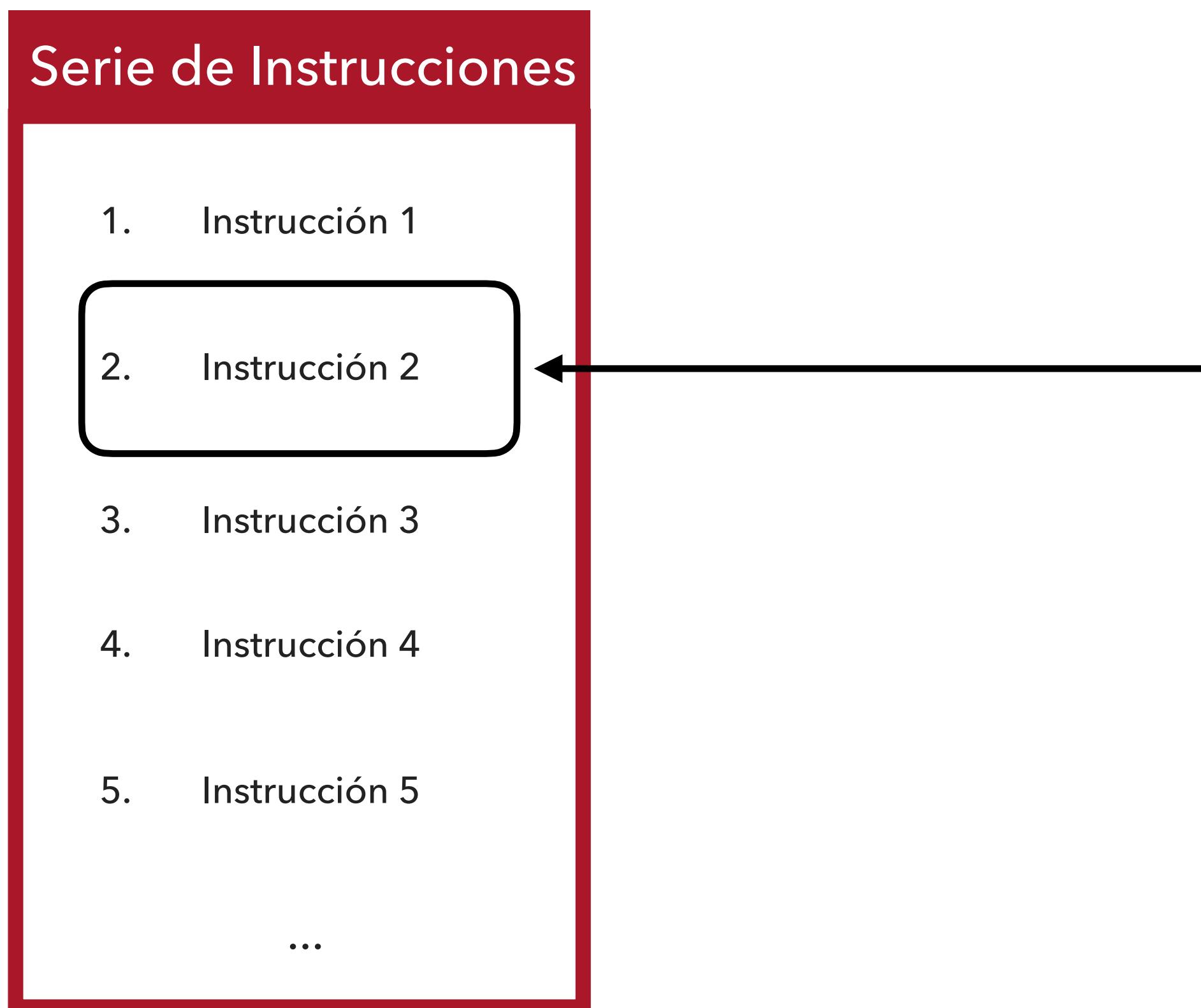
Funcionamiento de un intérprete:





¿CÓMO PASAMOS A CÓDIGO BINARIO?

Funcionamiento de un intérprete:

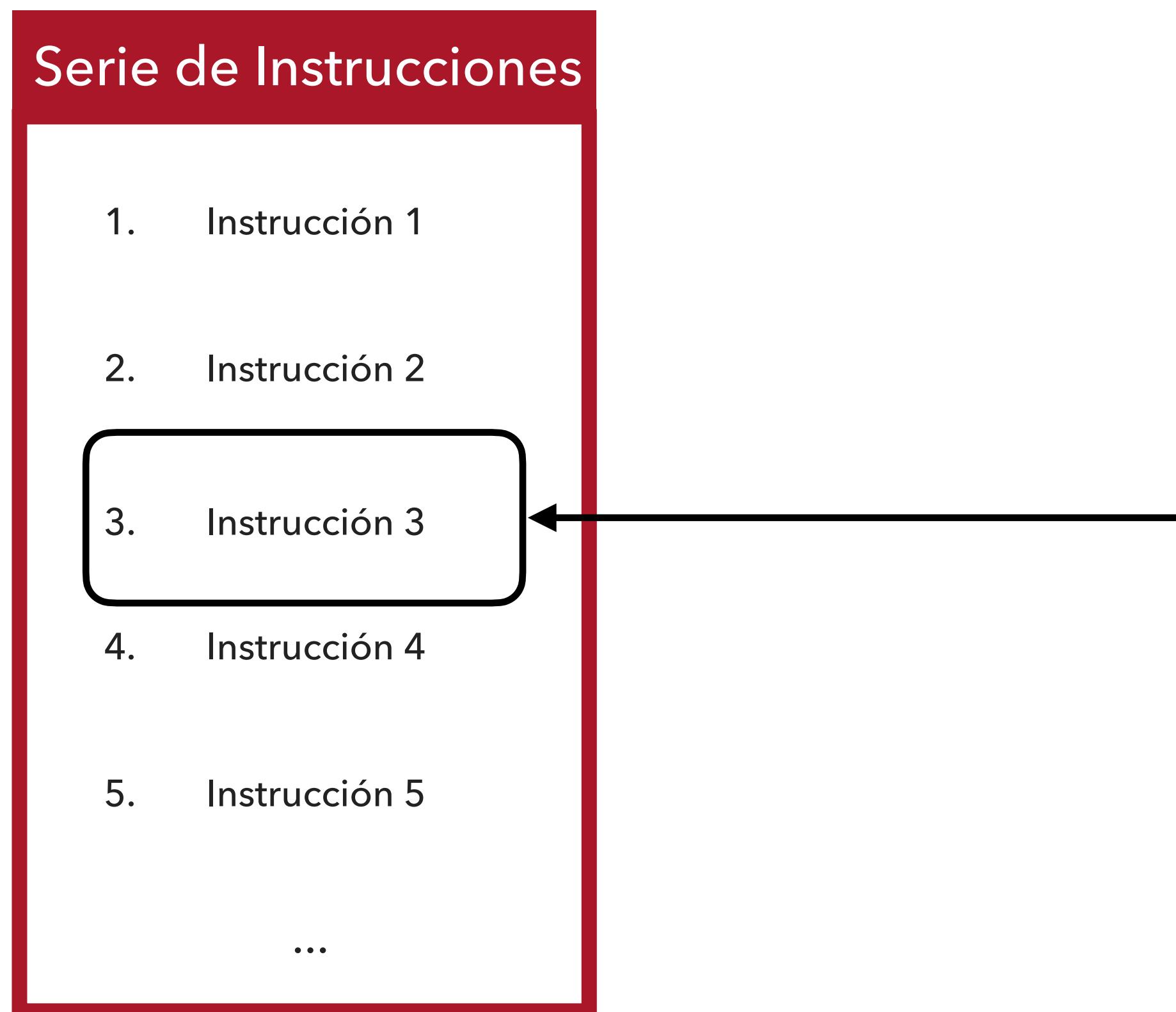


Realizar una traducción
↓
Ejecuta



¿CÓMO PASAMOS A CÓDIGO BINARIO?

Funcionamiento de un intérprete:

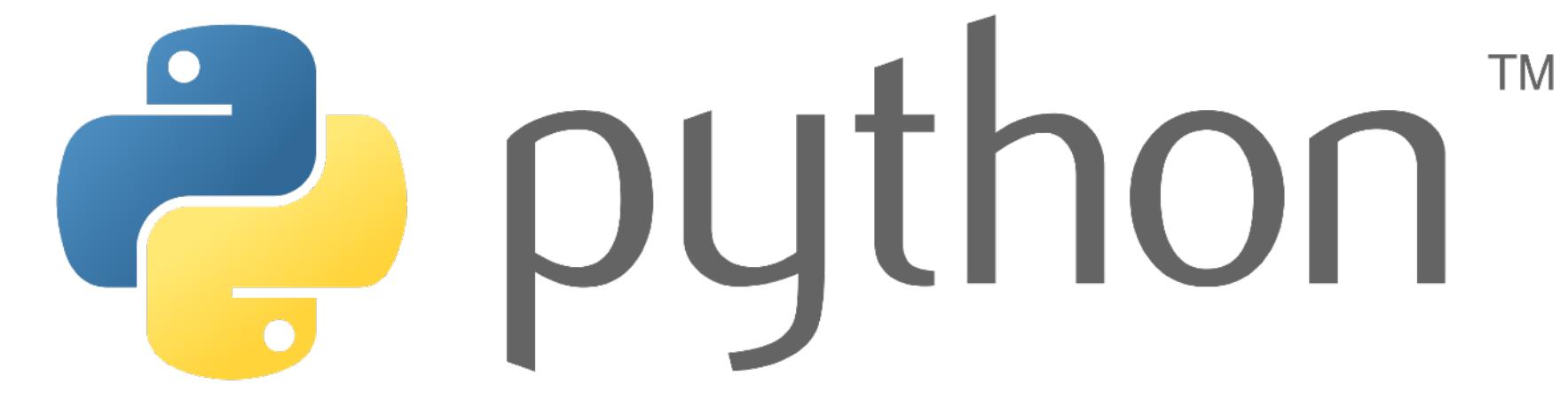
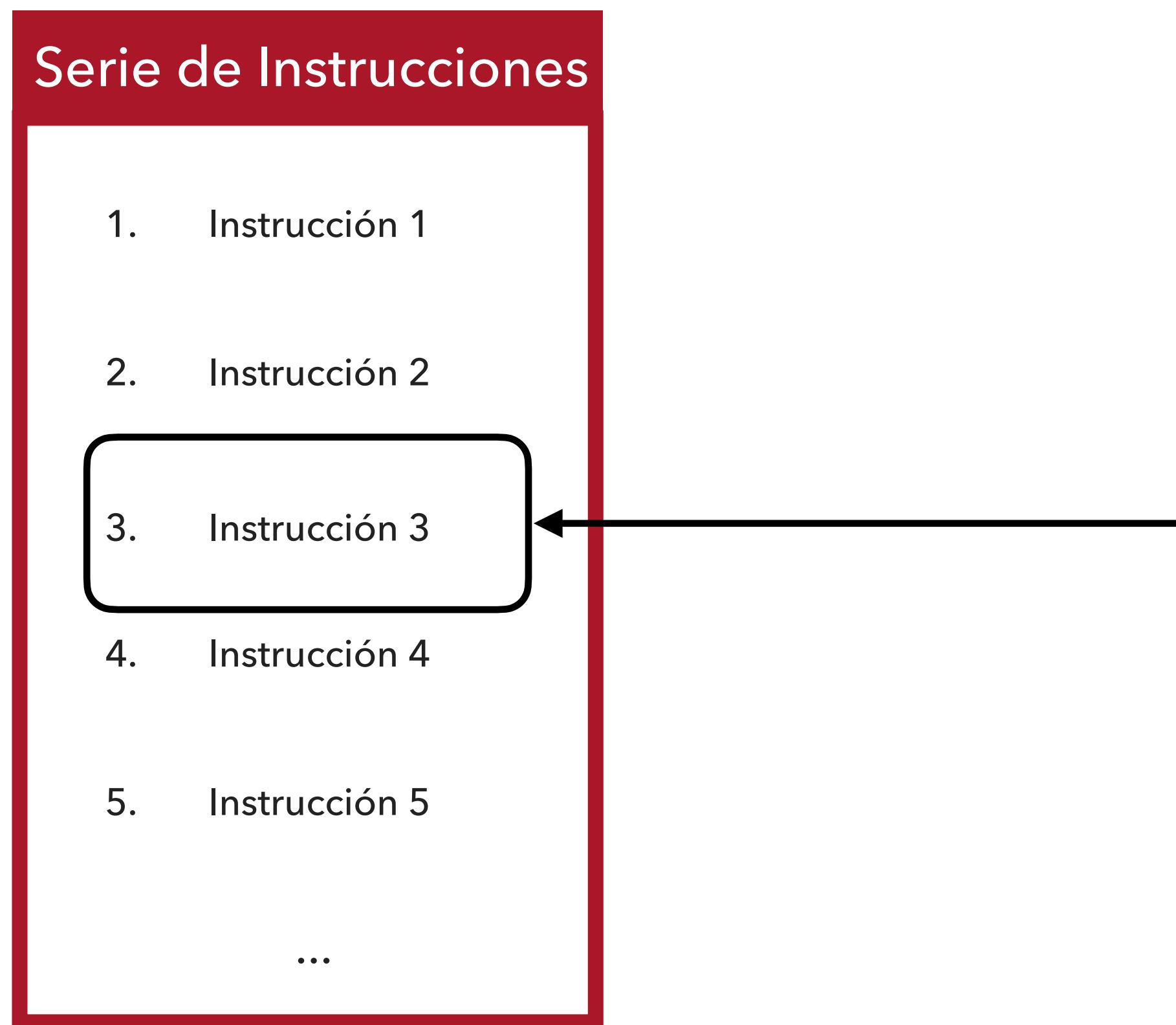


Realizar una traducción
↓
Ejecuta



¿CÓMO PASAMOS A CÓDIGO BINARIO?

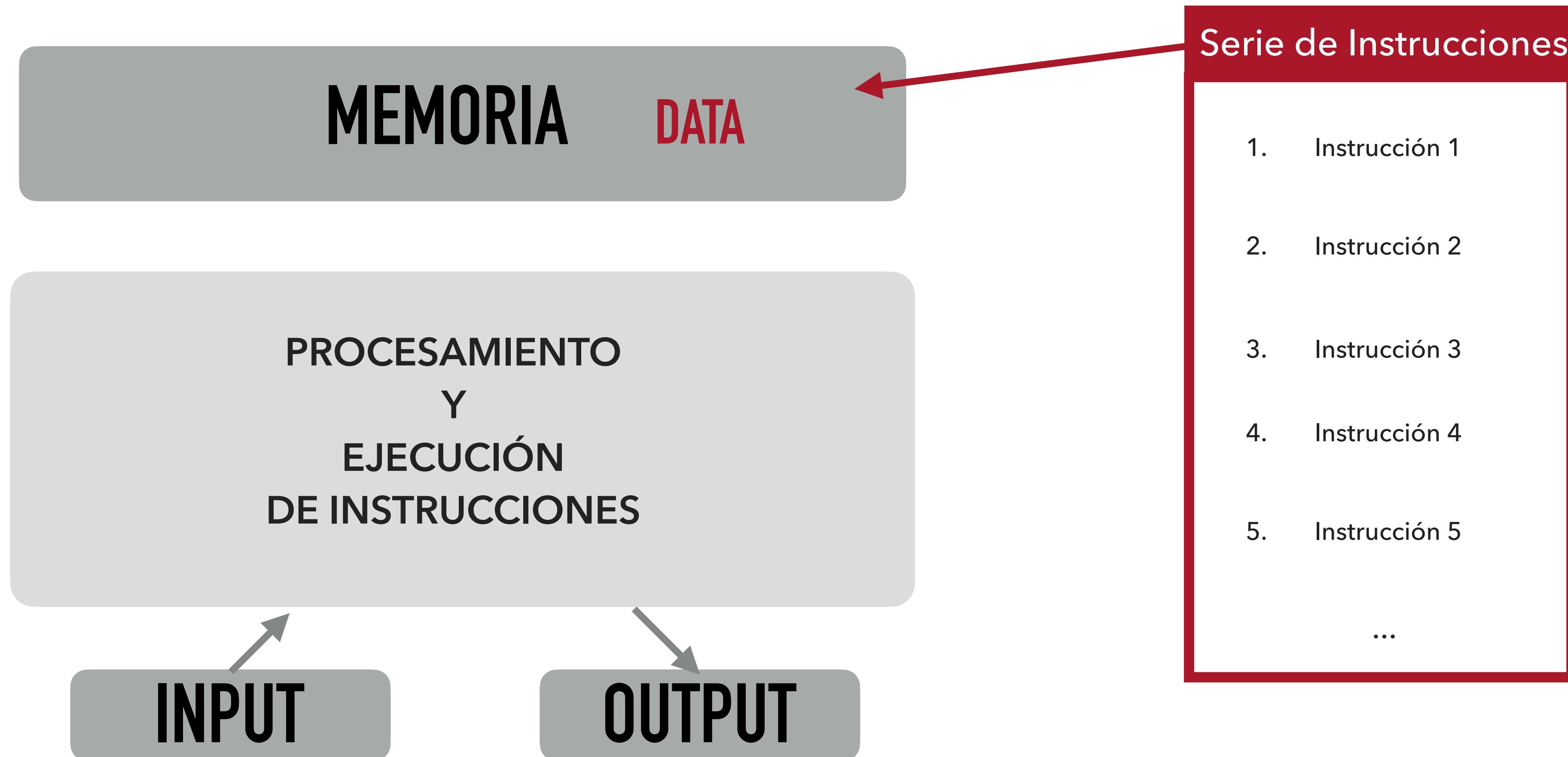
Funcionamiento de un intérprete:



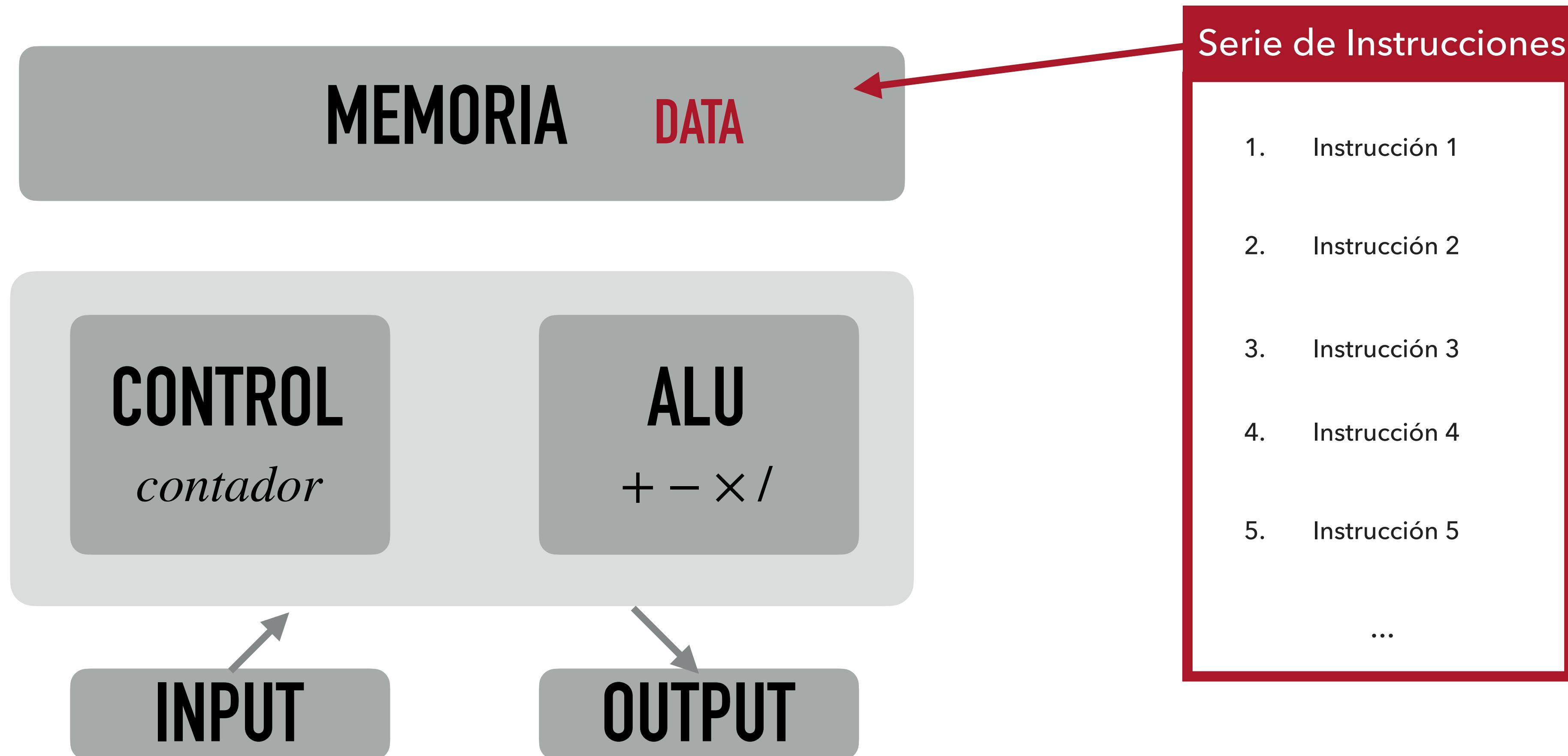
Realizar una traducción
↓
Ejecuta



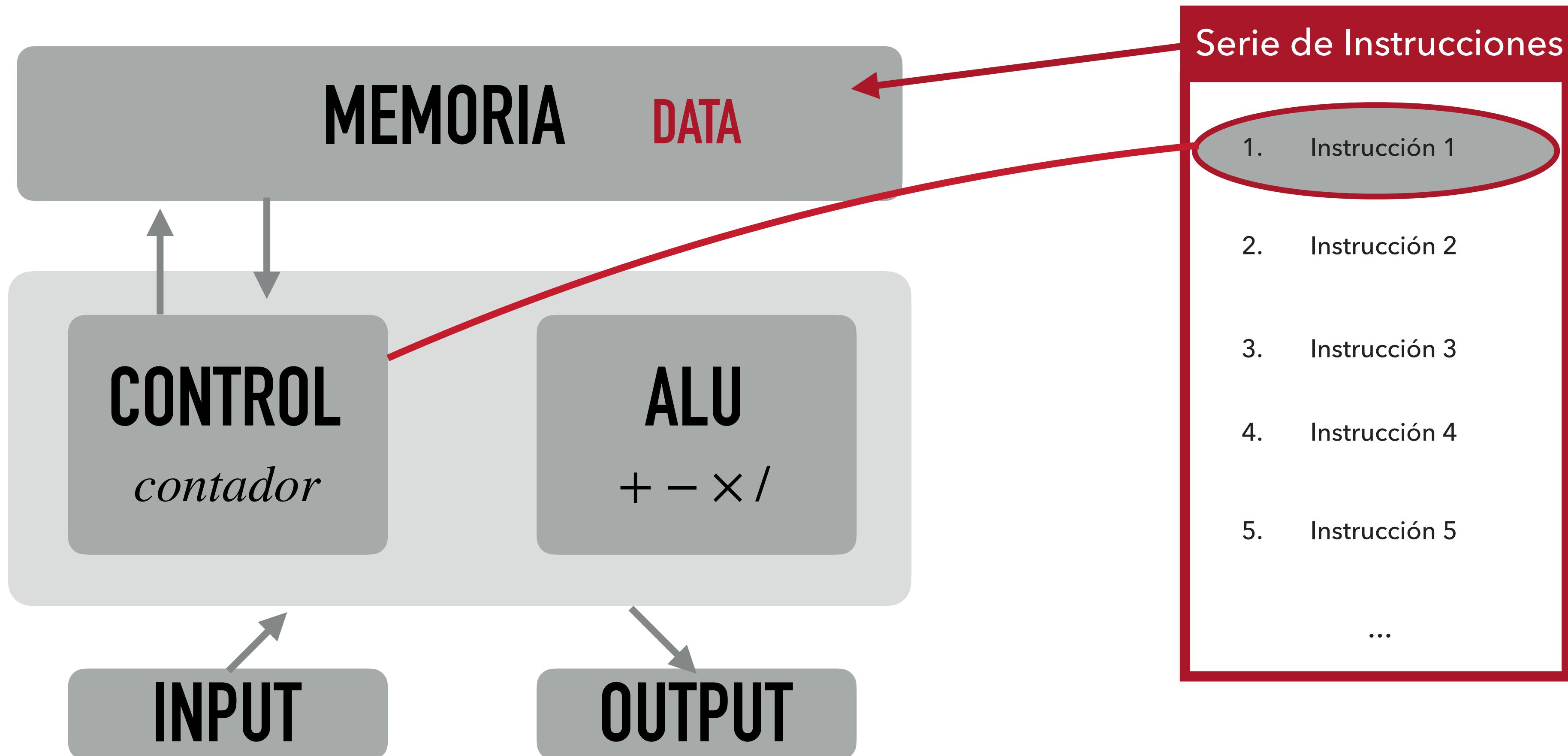
COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



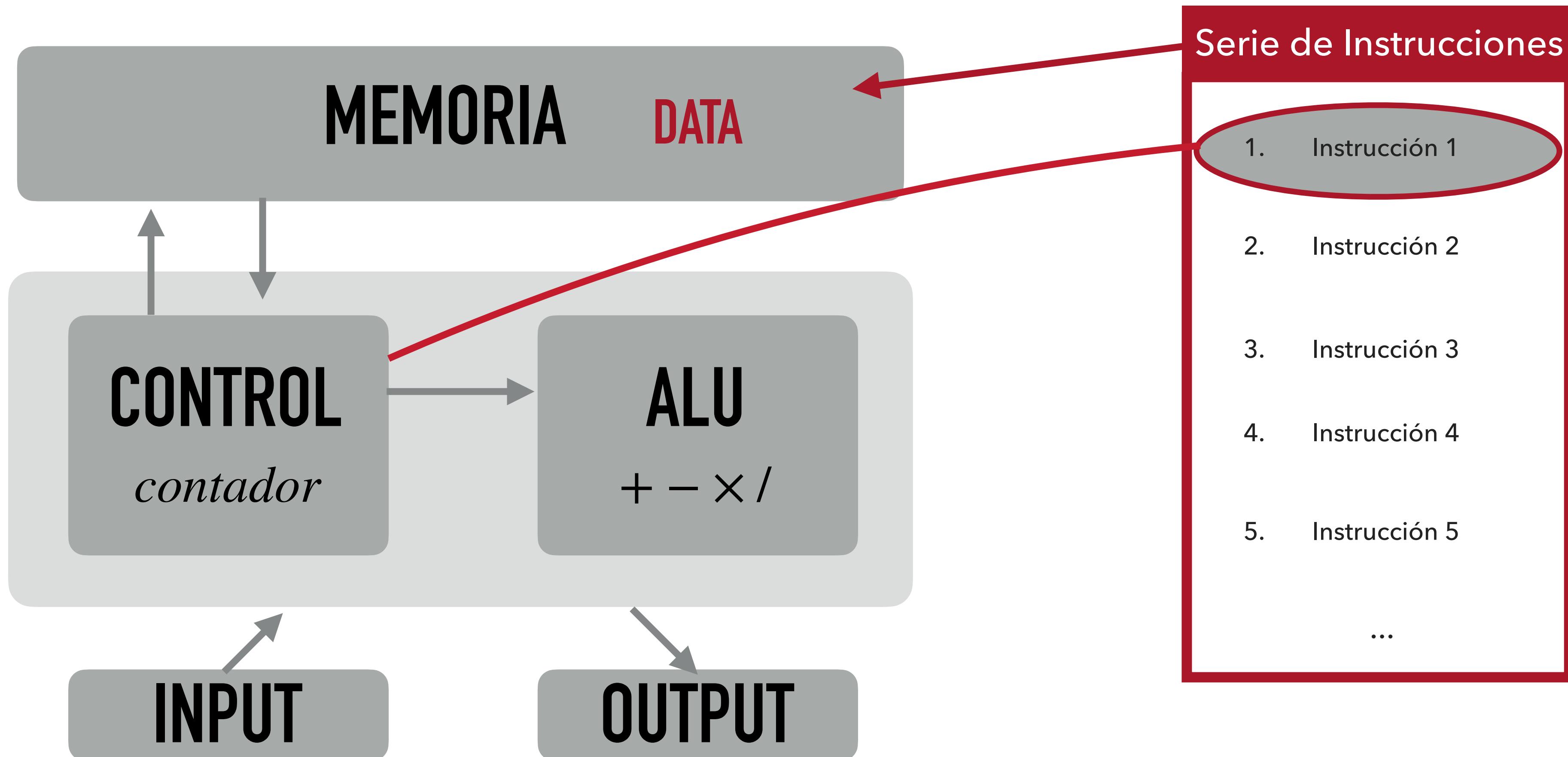
COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



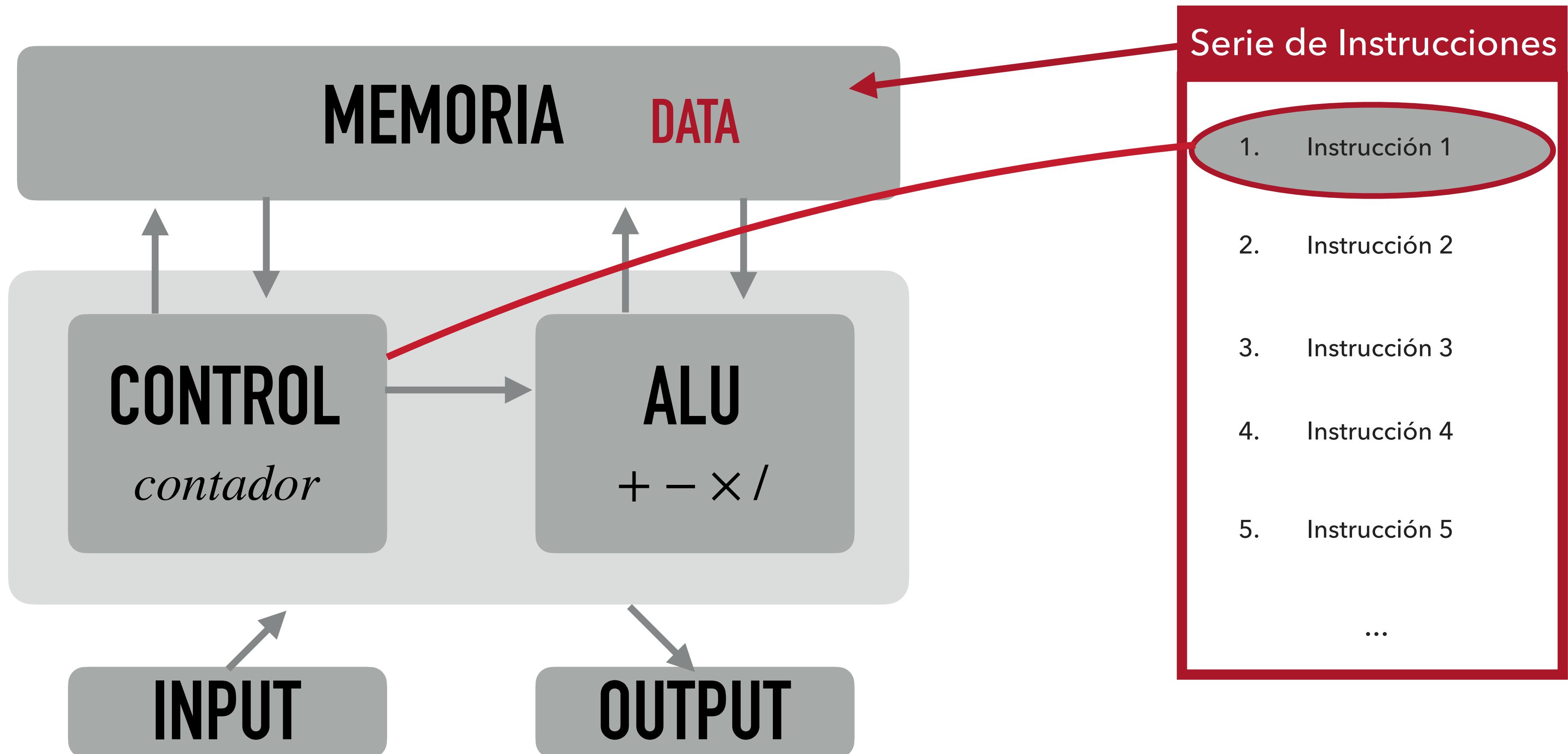
COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



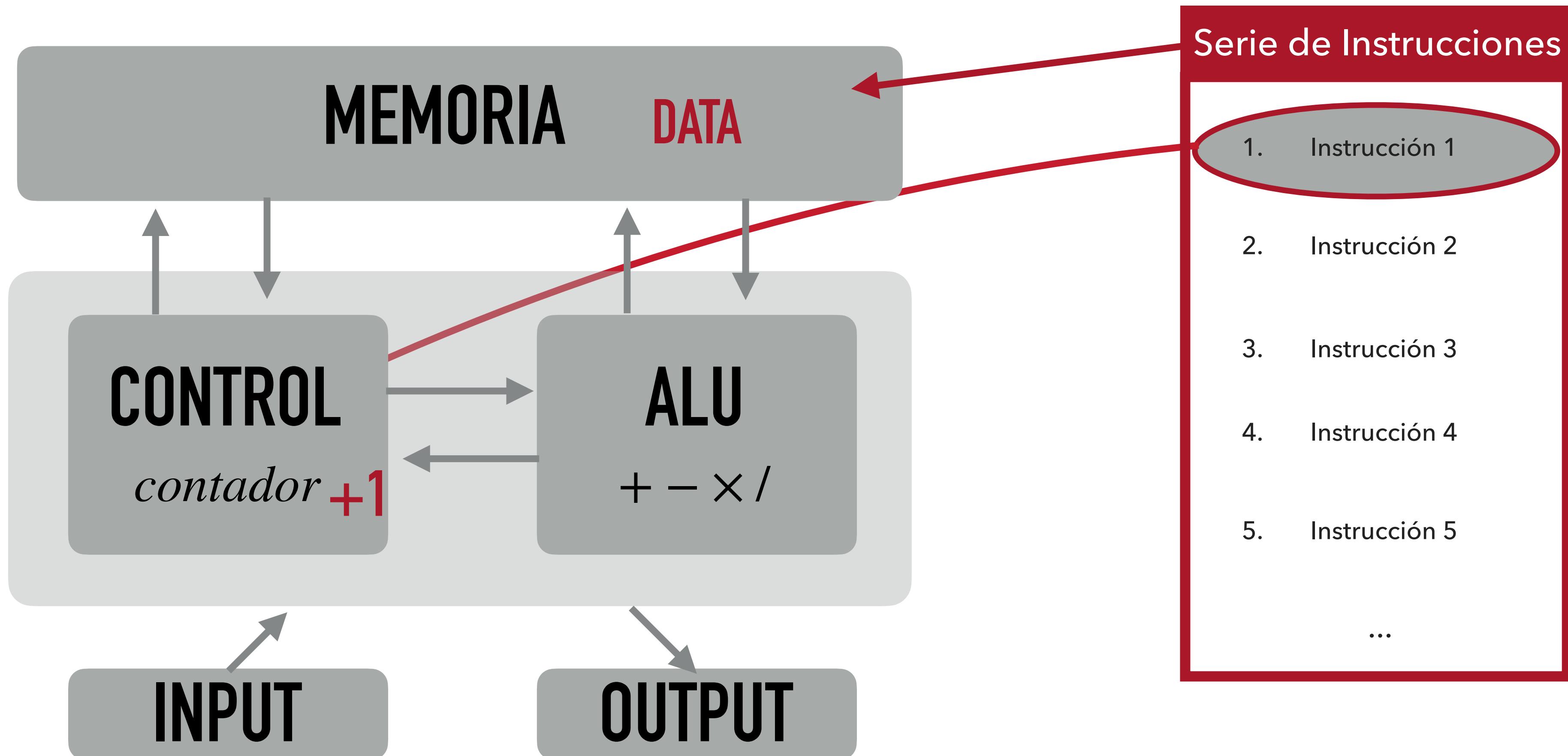
COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



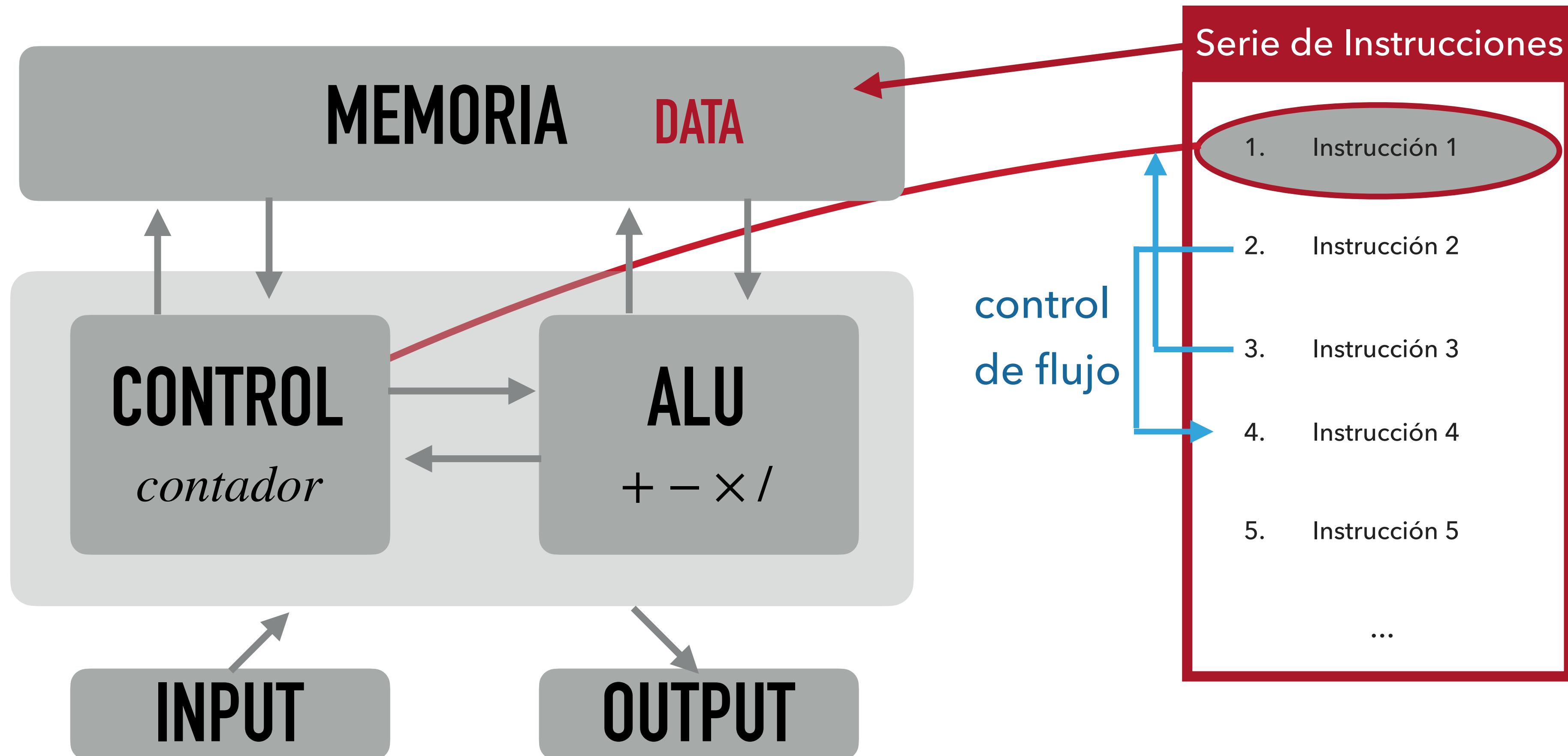
COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO

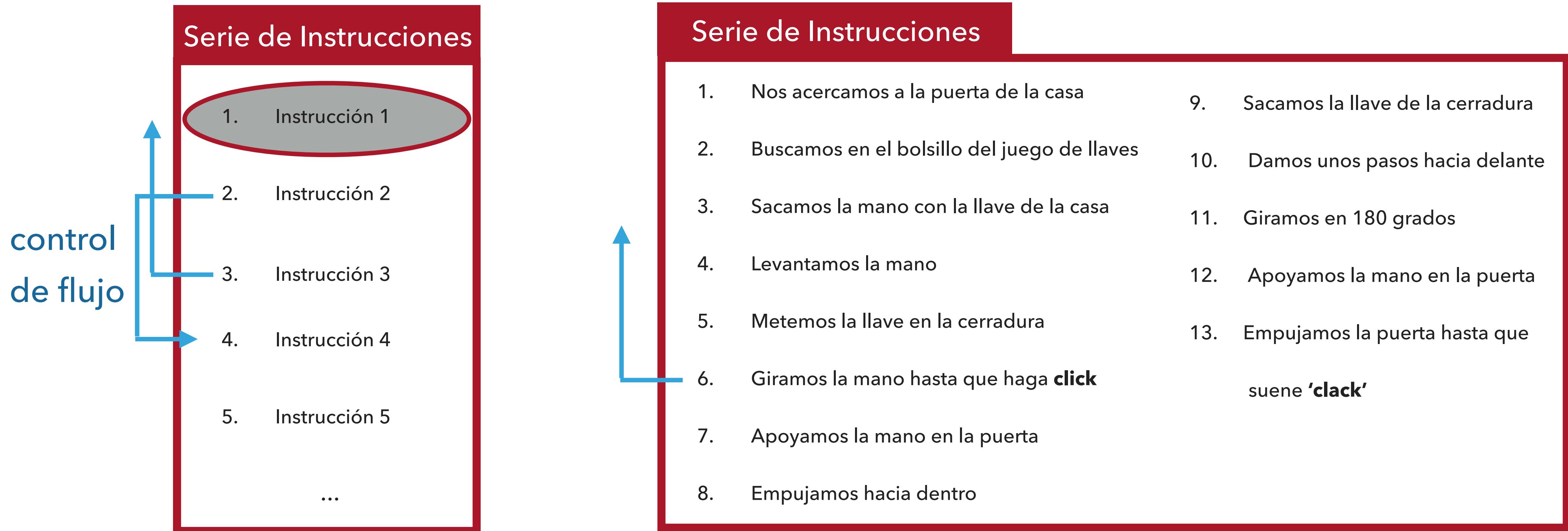


COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



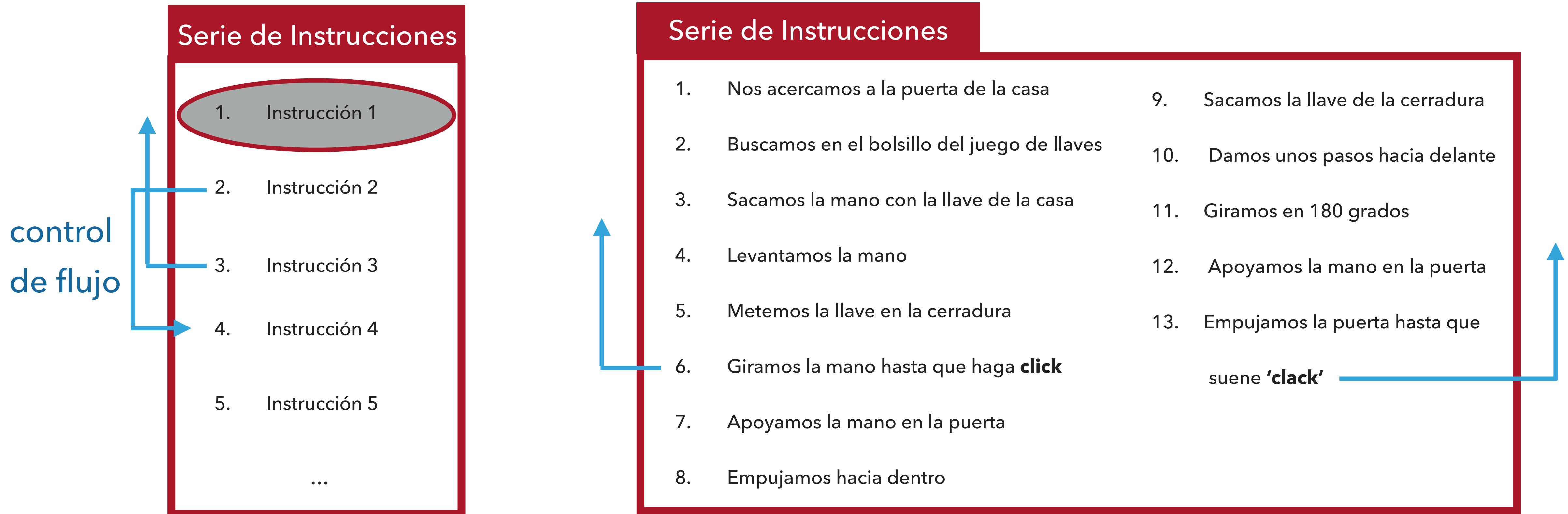


COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



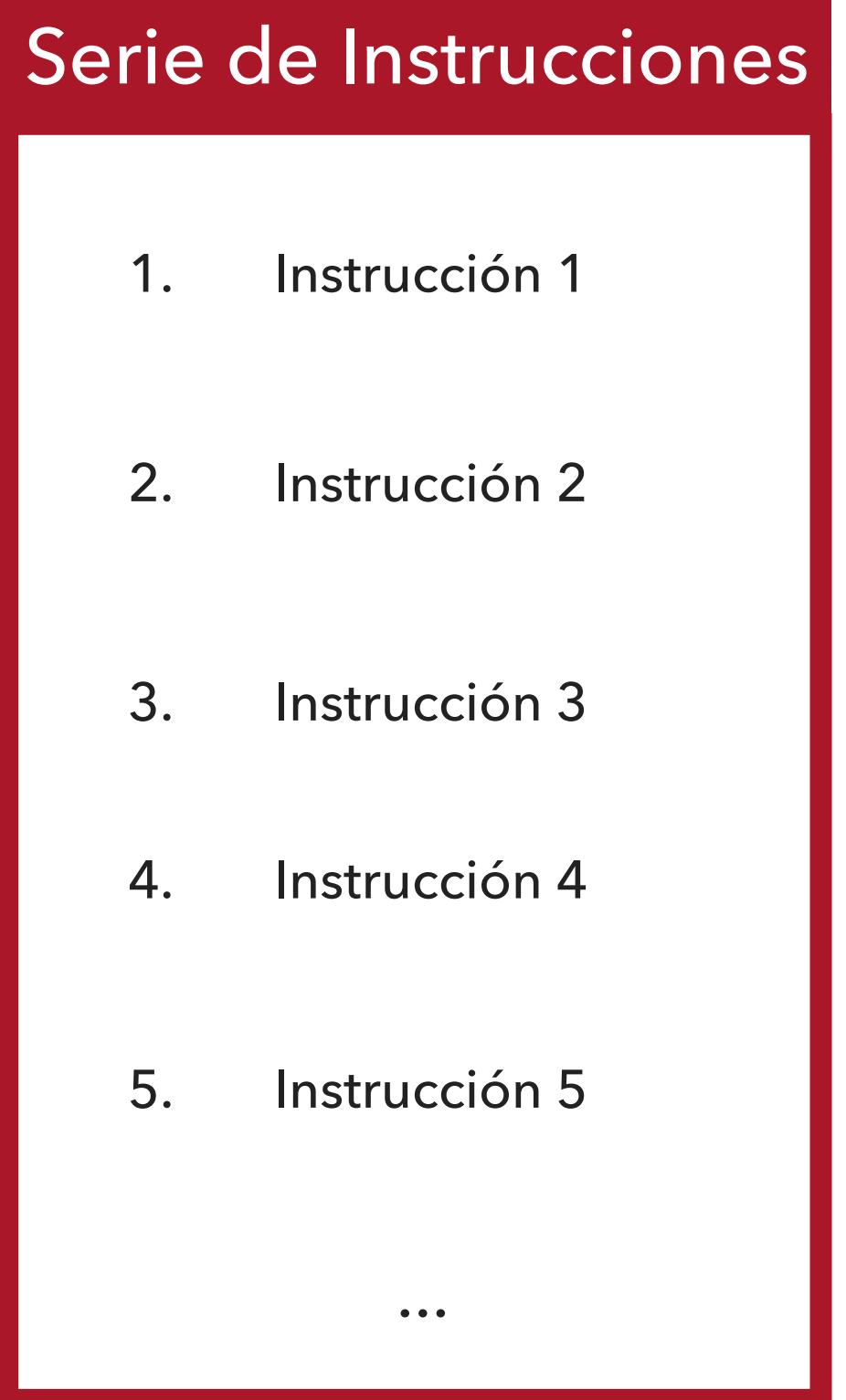


COMO INTERPRETA LA MAQUINA NUESTRO CÓDIGO



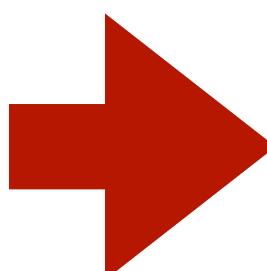
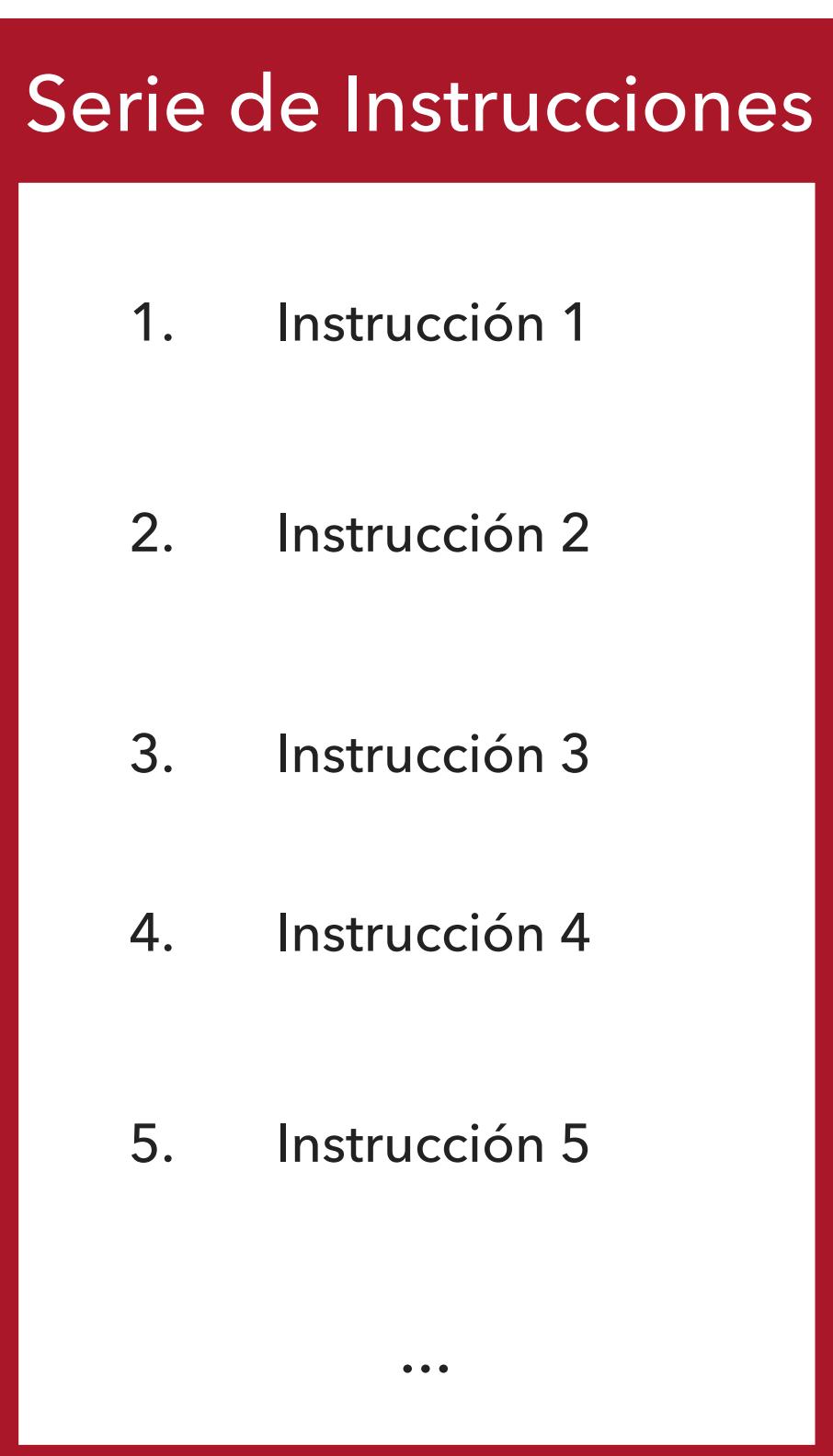


ALGORÍTMICA





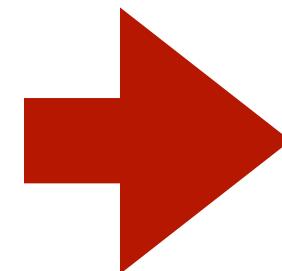
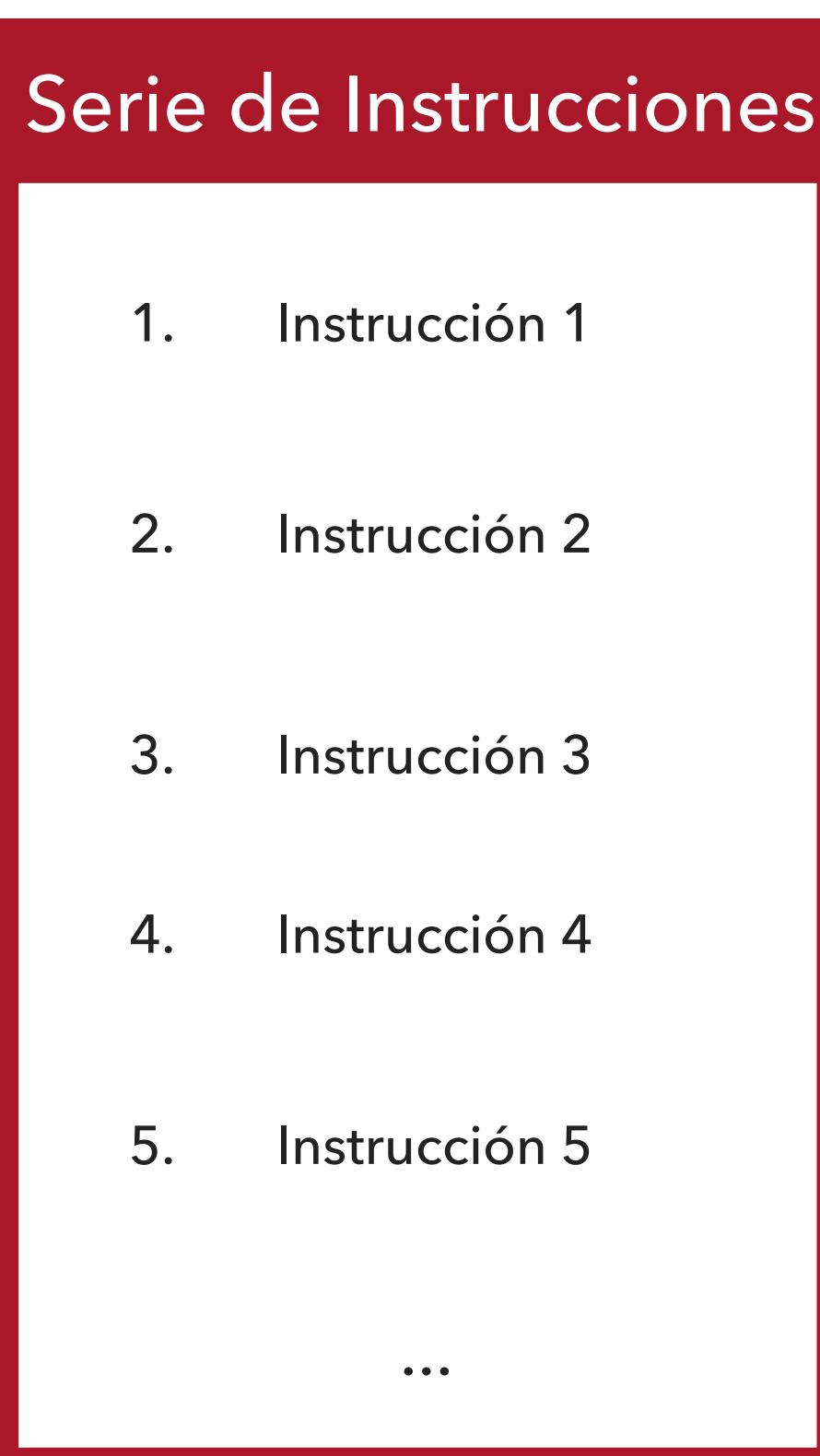
ALGORÍTMICA



ALGORÍTMO



ALGORÍTMICA

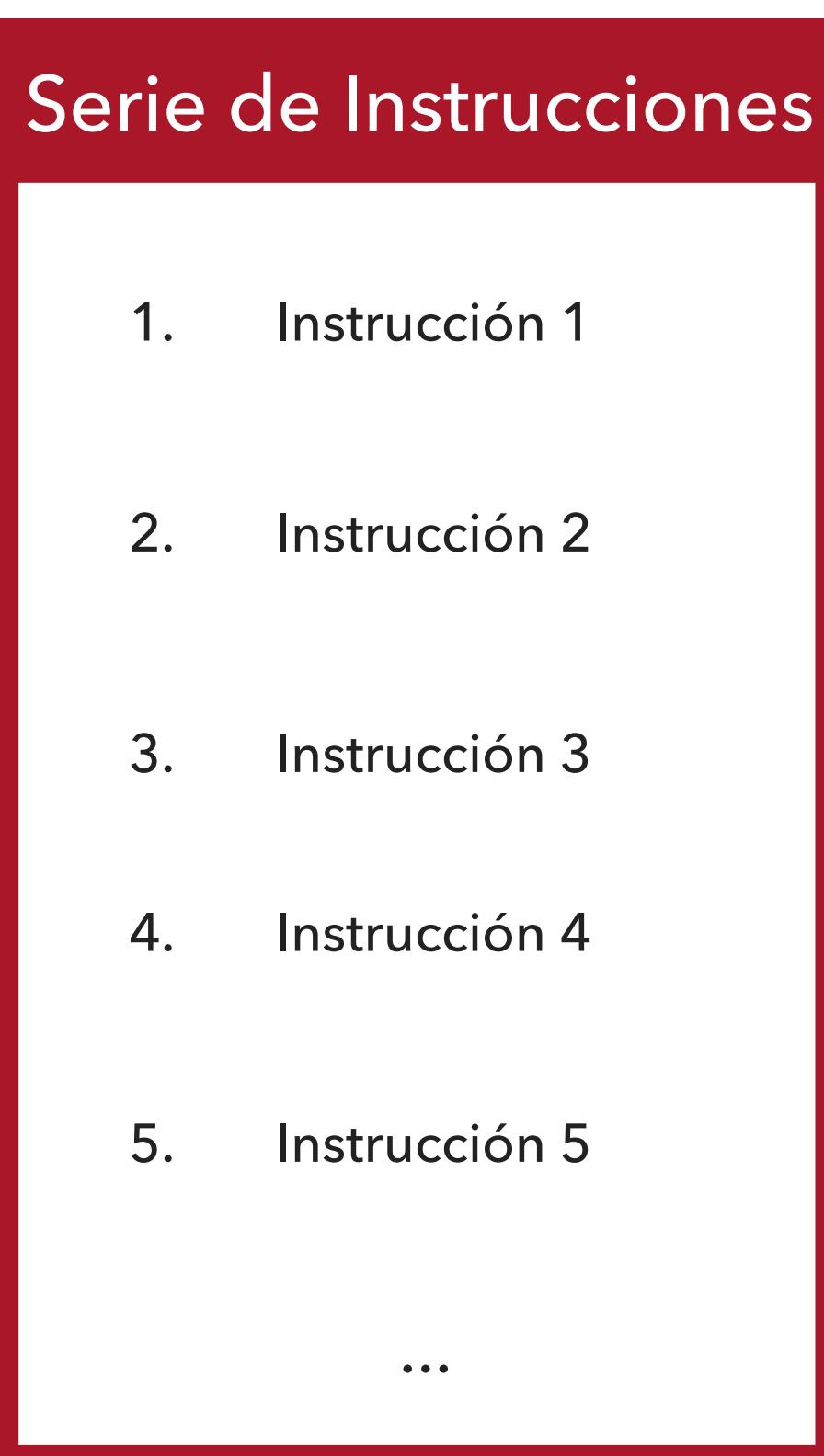


ALGORÍTMO

Conjunto bien definido de instrucciones que deben seguirse para la resolución de un problema



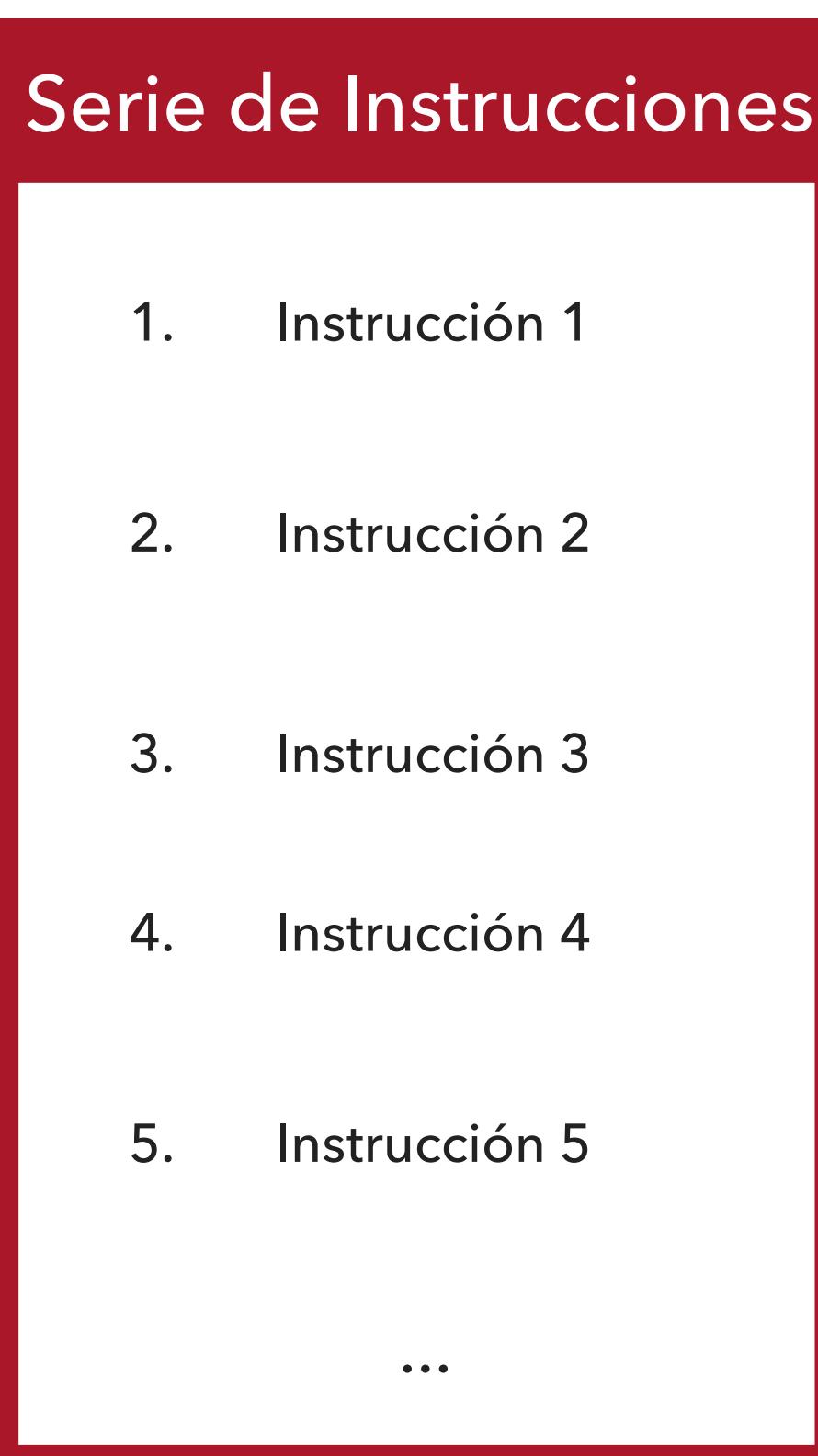
ALGORÍTMICA



ALGORÍTMO



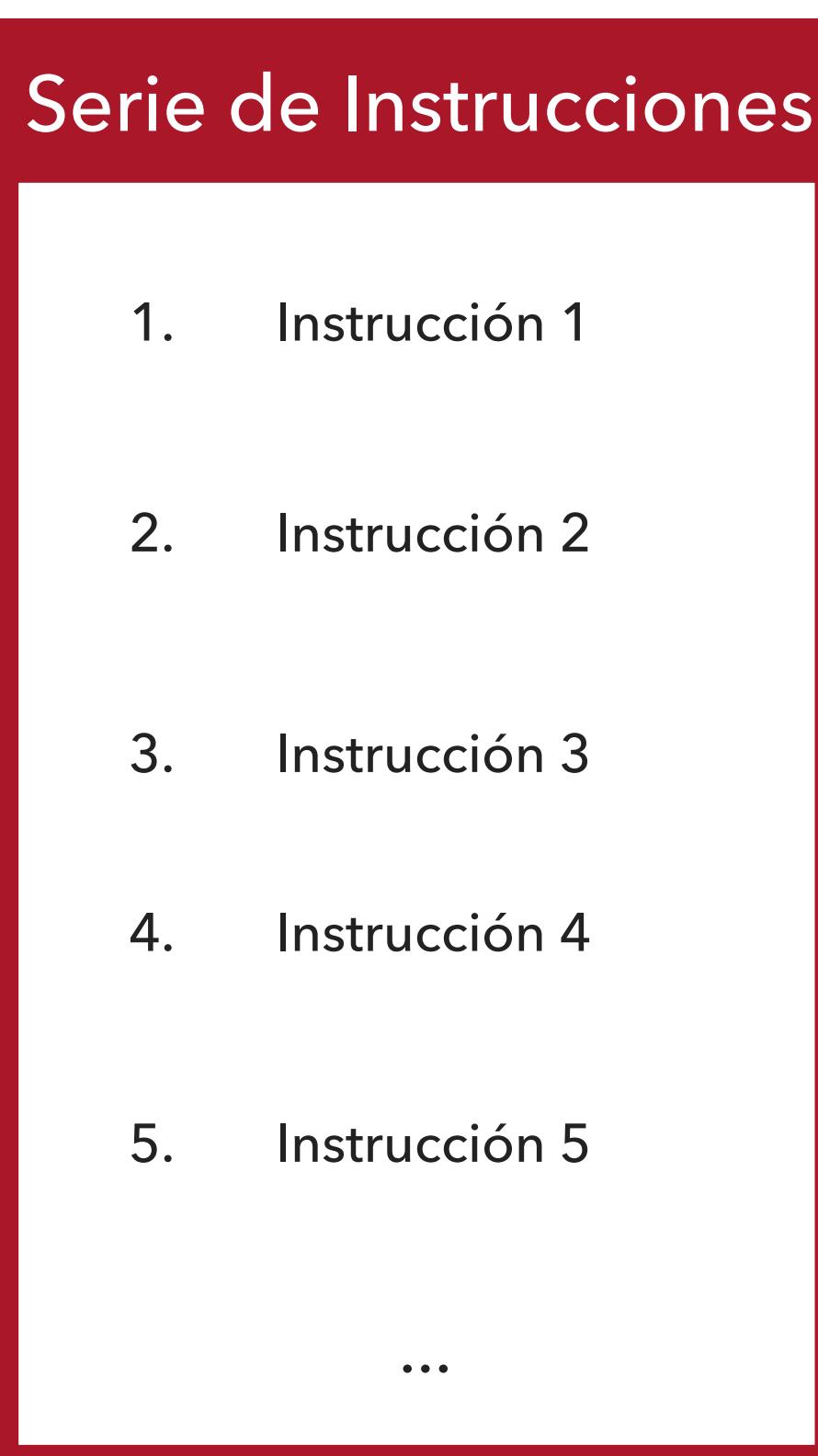
ALGORÍTMICA



ALGORÍTMO
+
ESTRUCTURAS DE DATOS



ALGORÍTMICA



ALGORÍTMO
+
ESTRUCTURAS DE DATOS
=
PROGRAMA



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?

- Lenguajes de programación



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?

- Lenguajes de programación
- Lenguaje natural



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?

- Lenguajes de programación
- Lenguaje natural
- Pseudocódigo
- Diagramas de flujo



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?

- Lenguajes de programación
- Lenguaje natural → descripciones extensas y ambiguas
- Pseudocódigo
- Diagramas de flujo



ALGORÍTMICA

¿CÓMO EXPRESAMOS UN ALGORITMO?

- Lenguajes de programación

- Lenguaje natural



descripciones extensas y ambiguas

- Pseudocódigo



- Diagramas de flujo

evitan ambigüedades

formas estructuradas de expresar algoritmos

independientes de un lenguaje de
programación específico



ALGORÍTMICA

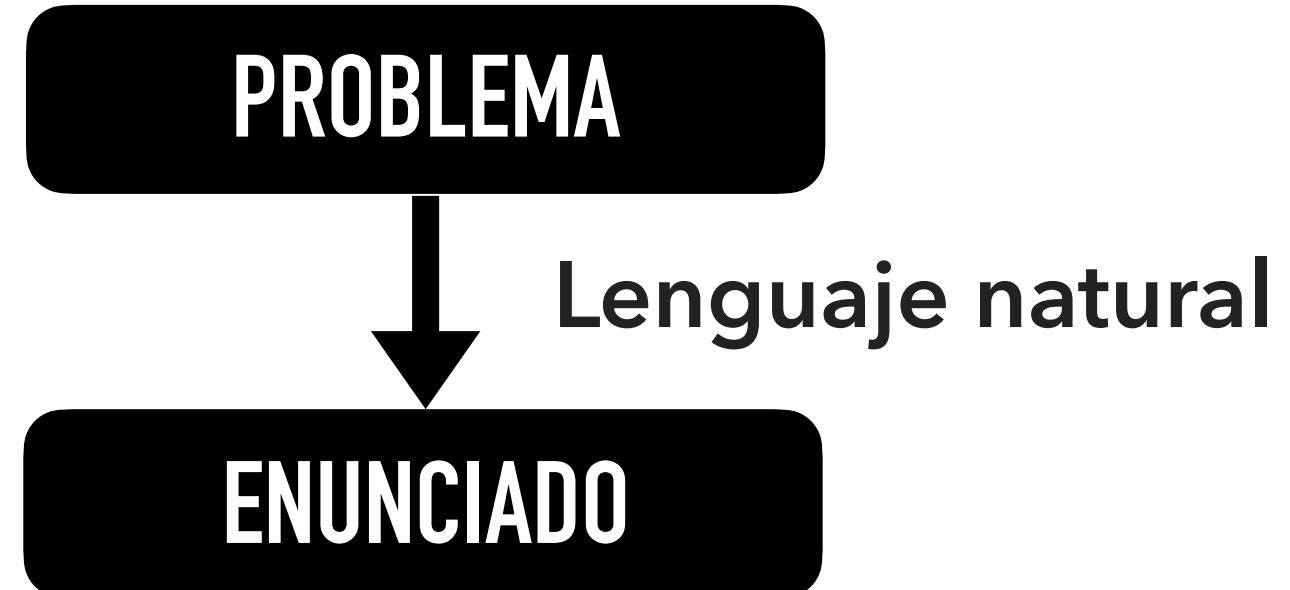
CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO

PROBLEMA



ALGORÍTMICA

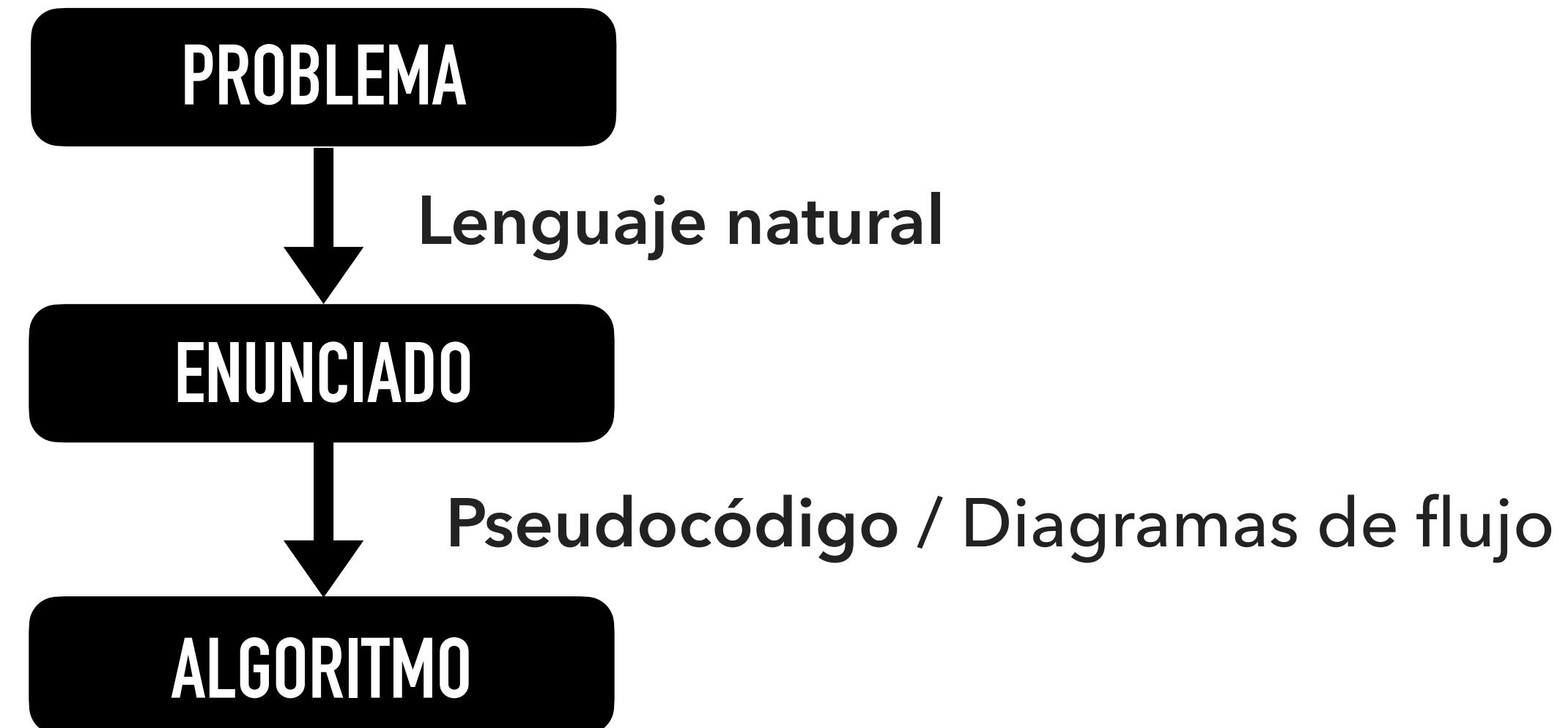
CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO





ALGORÍTMICA

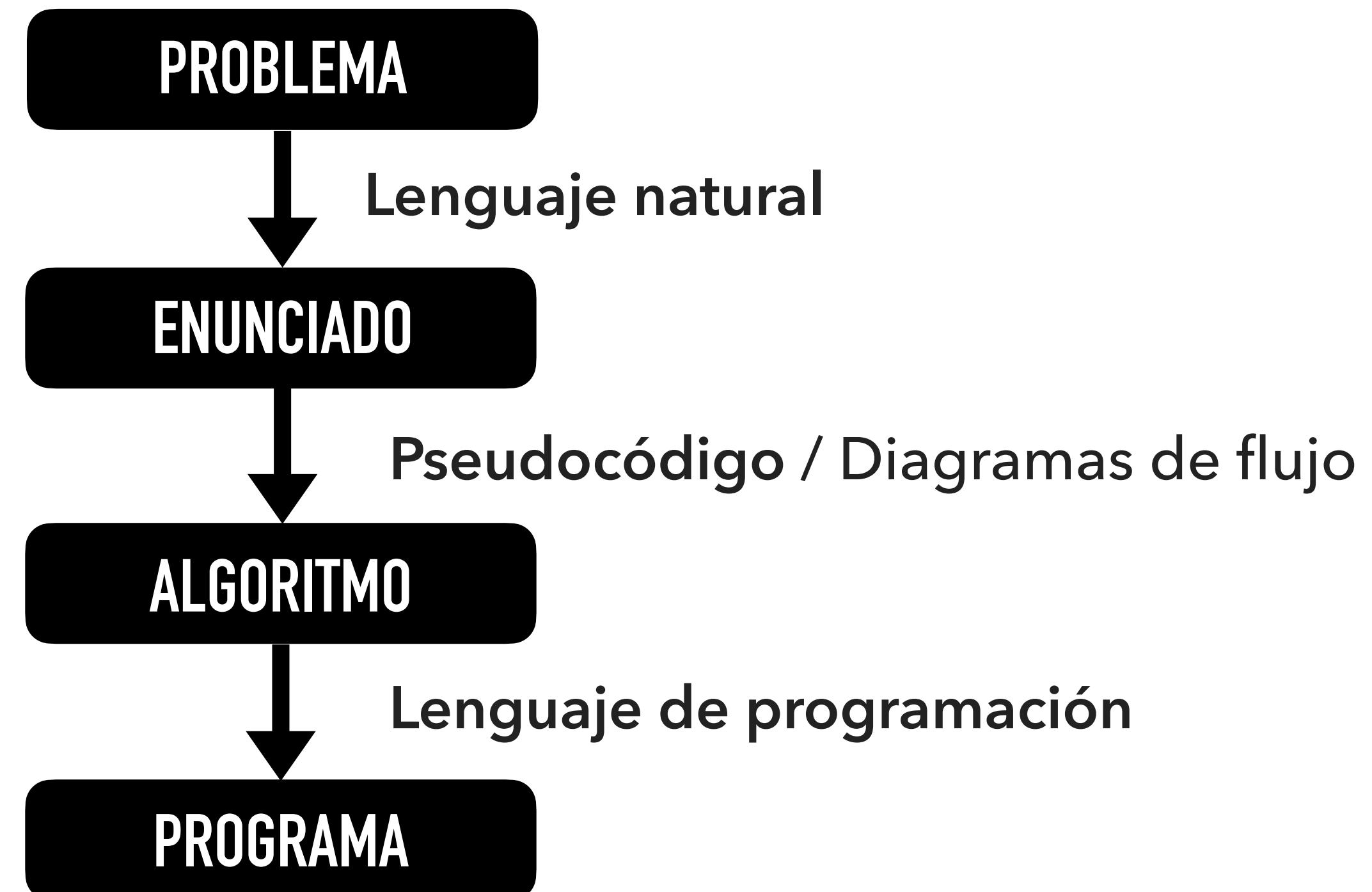
CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO





ALGORÍTMICA

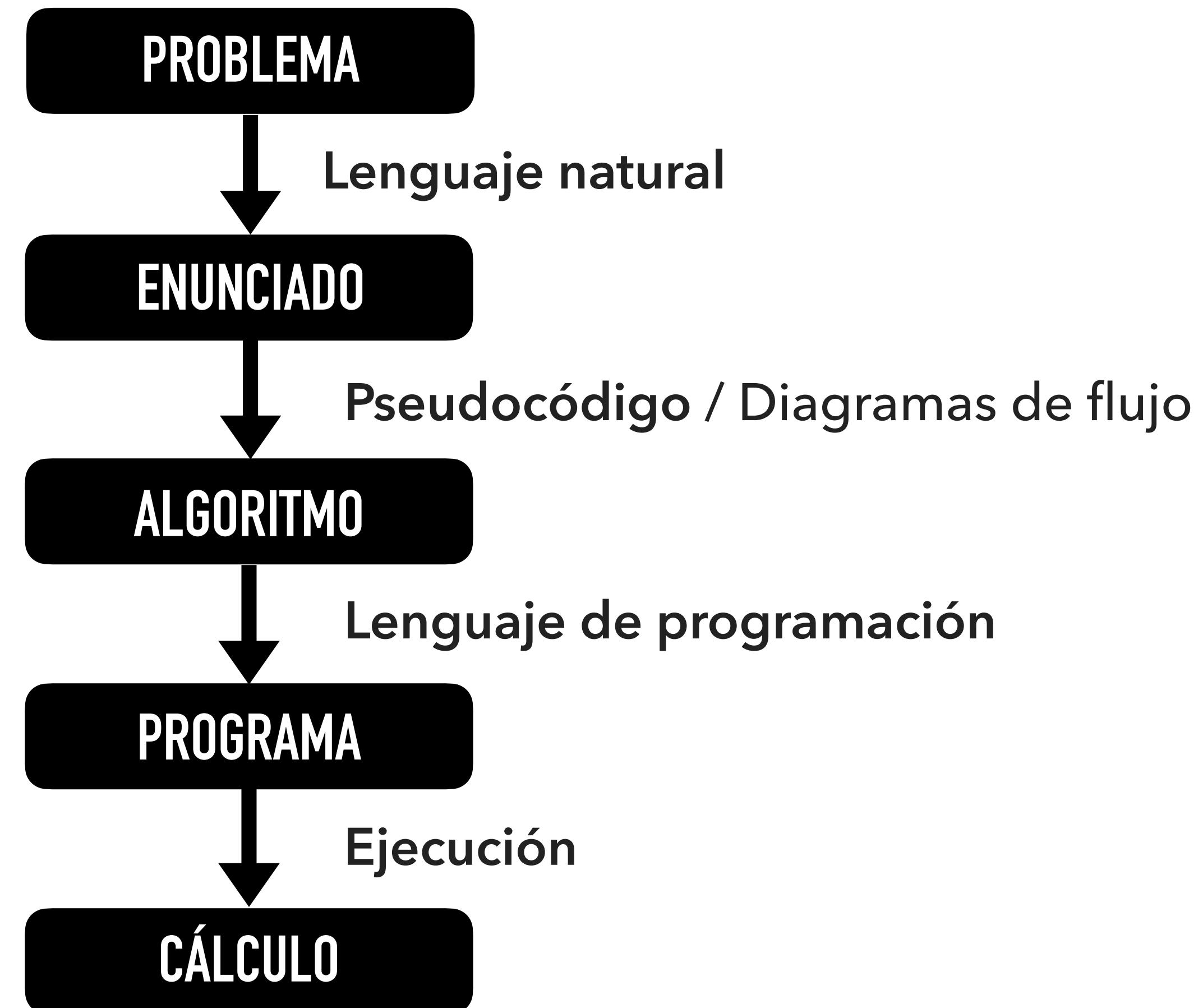
CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO





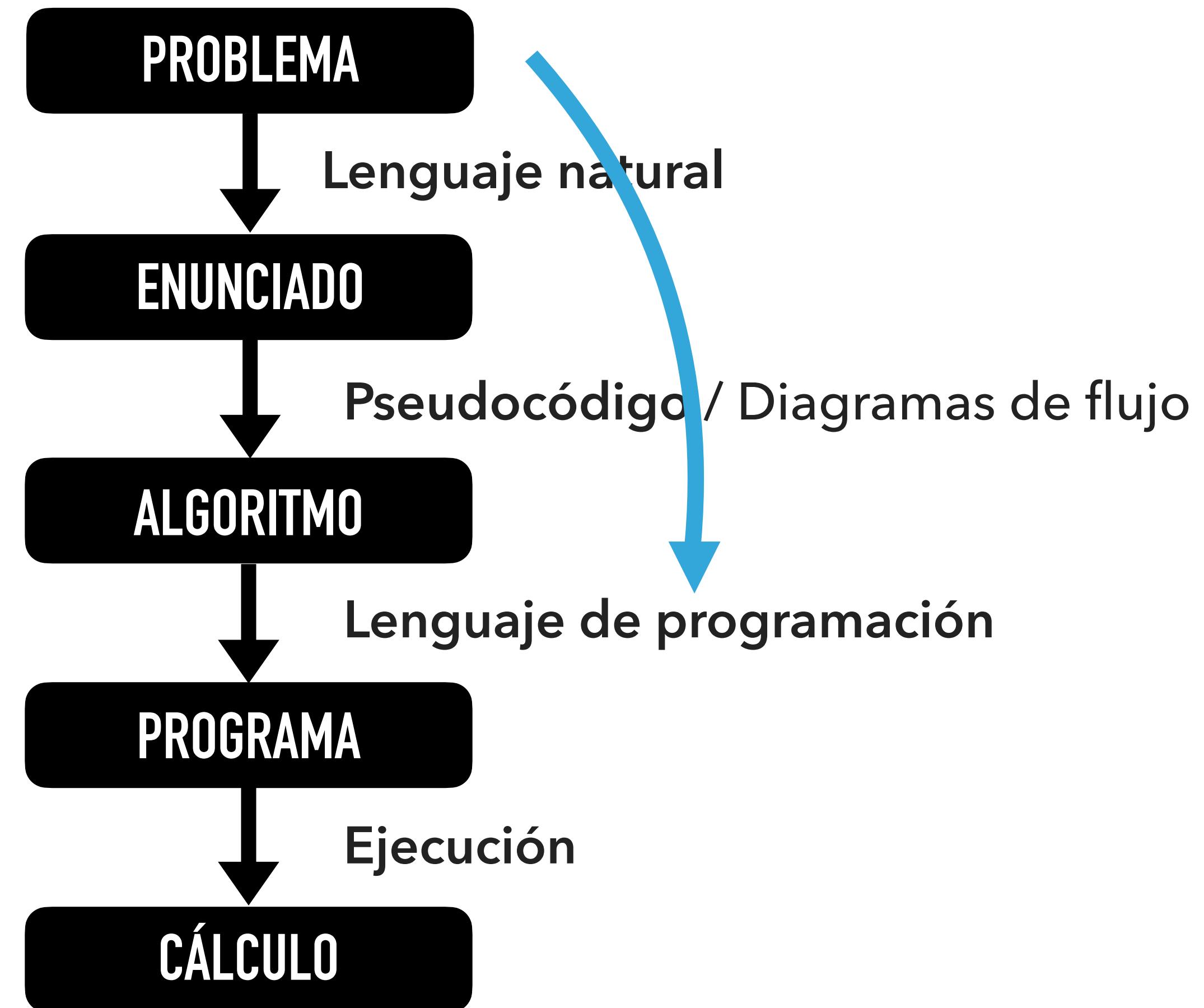
ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: WORKFLOW O FLUJO DE TRABAJO





ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ...

INTEGER - ENTERO

2.7, 12.6...

FLOAT - DECIMAL



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ...

INTEGER - ENTERO

2.7, 12.6...

FLOAT - DECIMAL

2. Caractéres

"abc"

STRING - TEXTO



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ...

INTEGER - ENTERO

2.7, 12.6...

FLOAT - DECIMAL

2. Caractéres

"abc"

STRING - TEXTO

3. Lógicos

True / False

BOOLEAN



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ...	INTEGER - ENTERO
2.7, 12.6...	FLOAT - DECIMAL

2. Caractéres

"abc"	STRING - TEXTO
-------	----------------

3. Lógicos

True / False	BOOLEAN
--------------	---------

Operaciones primitivas:

- Suma
- Resta
- Multiplicación
- División
- Exponenciación
- Resto (de una división)



ALGORÍTMICA

CÓMO CREAMOS UN ALGORITMO: ESTRUCTURA

Datos:

1. Números

2, 7, 12 ... INTEGER - ENTERO

2.7, 12.6... FLOAT - DECIMAL

2. Caractéres

"abc" STRING - TEXTO

3. Lógicos

True / False BOOLEAN

Operaciones primitivas:

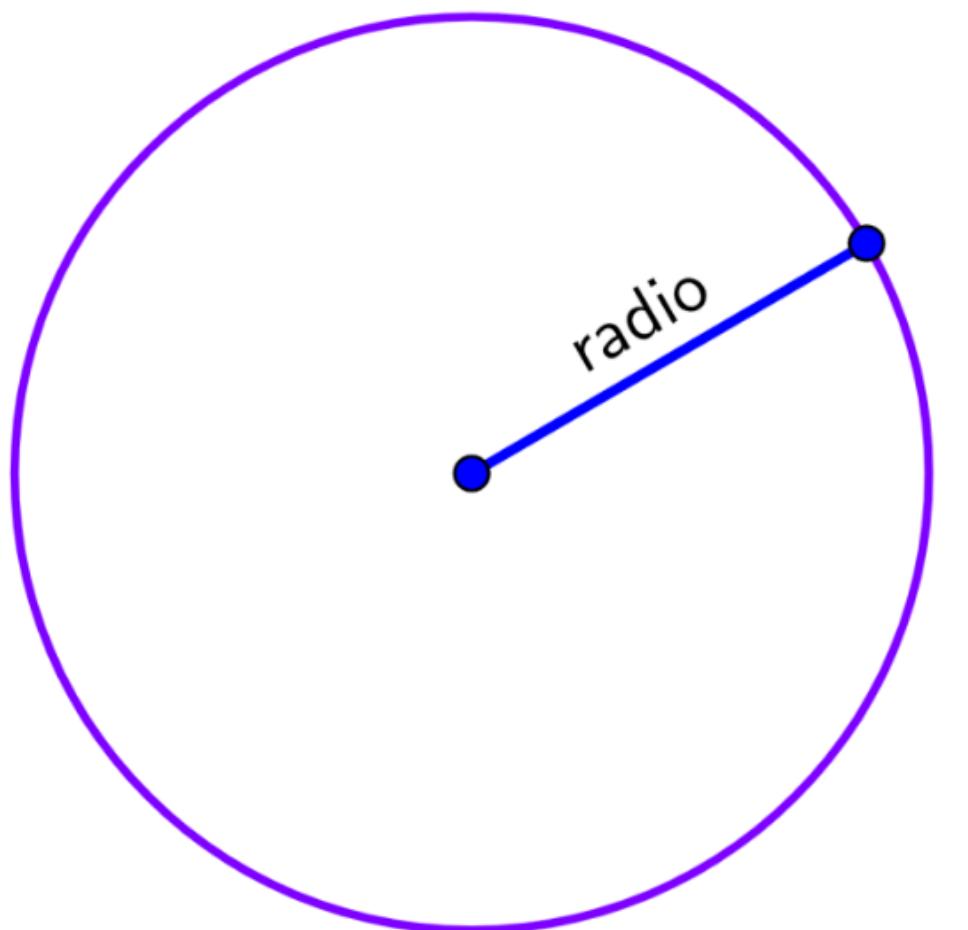
- Suma
- Resta
- Multiplicación
- División
- Exponenciación
- Resto (de una división)

Asignaciones:

- Variables
- Constantes
- Expresiones



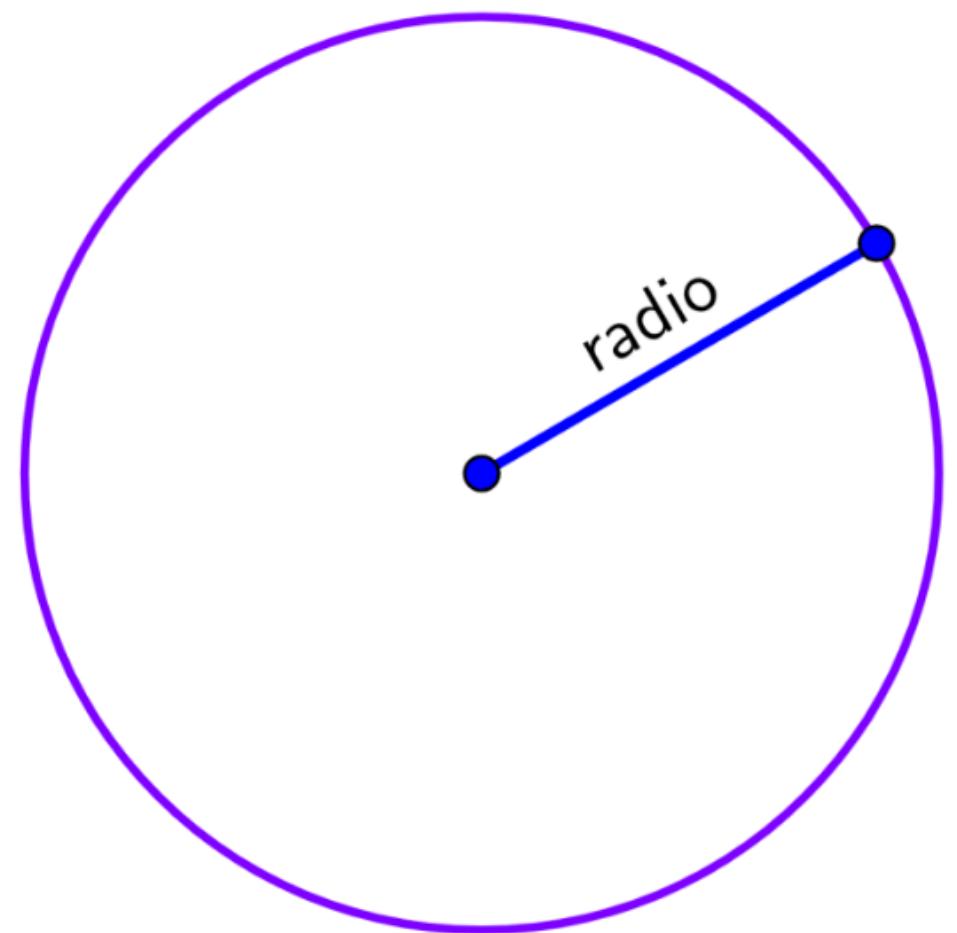
EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$



EJEMPLO: AREA DE UN CÍRCULO



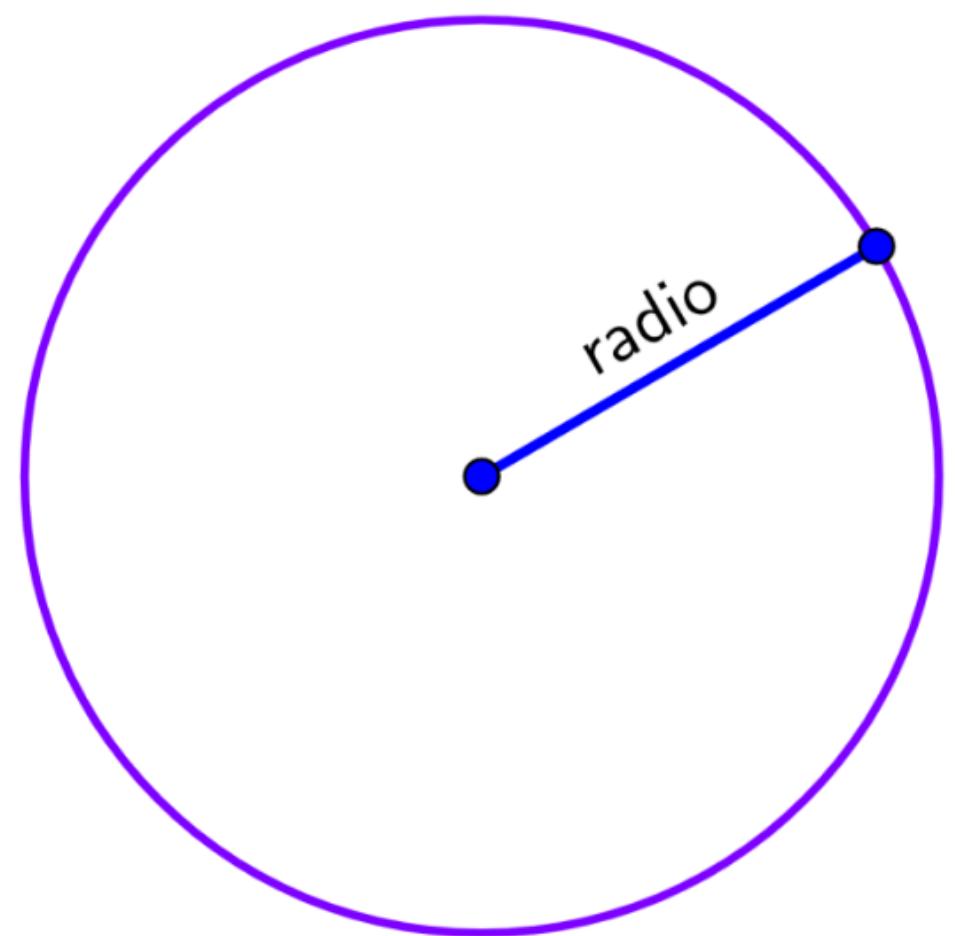
$$\text{Pi} = 3.141 \longrightarrow$$

creamos un vínculo entre la palabra
Pi y el valor 3.141

$$A = \pi r^2$$



EJEMPLO: AREA DE UN CÍRCULO



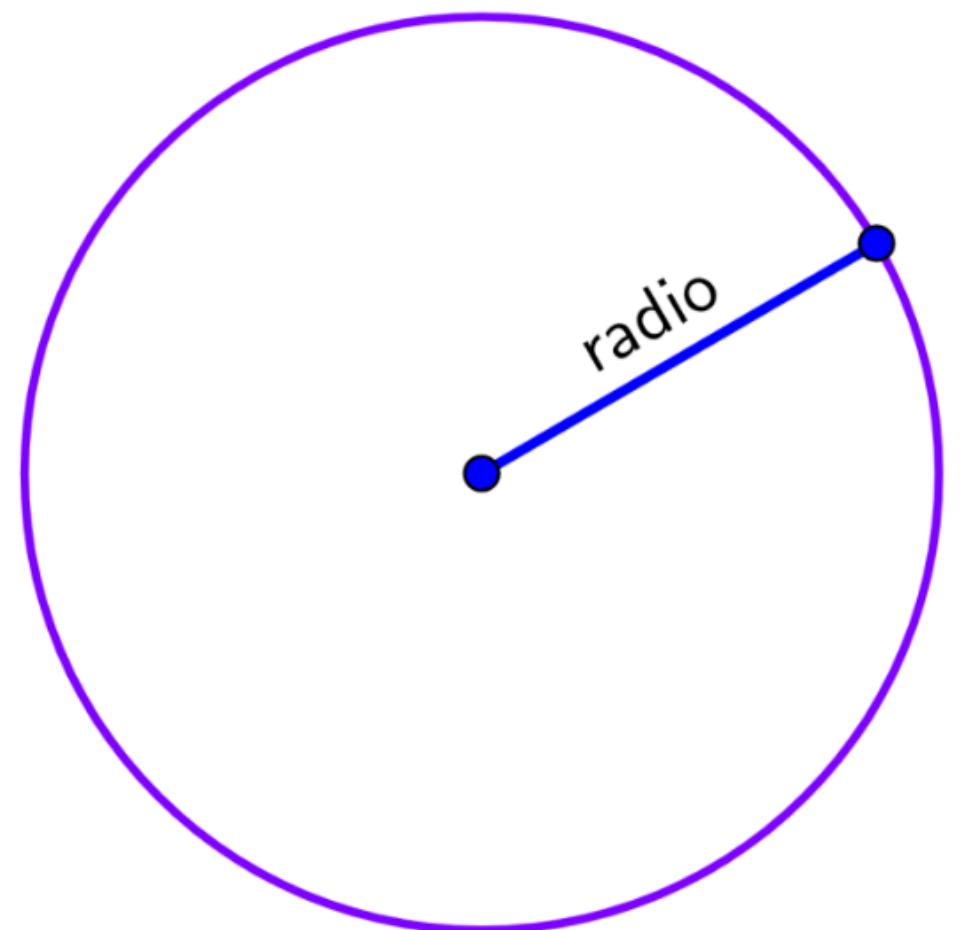
$$A = \pi r^2$$

$$\text{Pi} = 3.141 \longrightarrow$$

$$r = 1.5$$

creamos un vínculo entre la palabra
Pi y el valor 3.141

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

$$\text{Pi} = 3.141 \longrightarrow$$

$$\text{radio} = 1.5$$

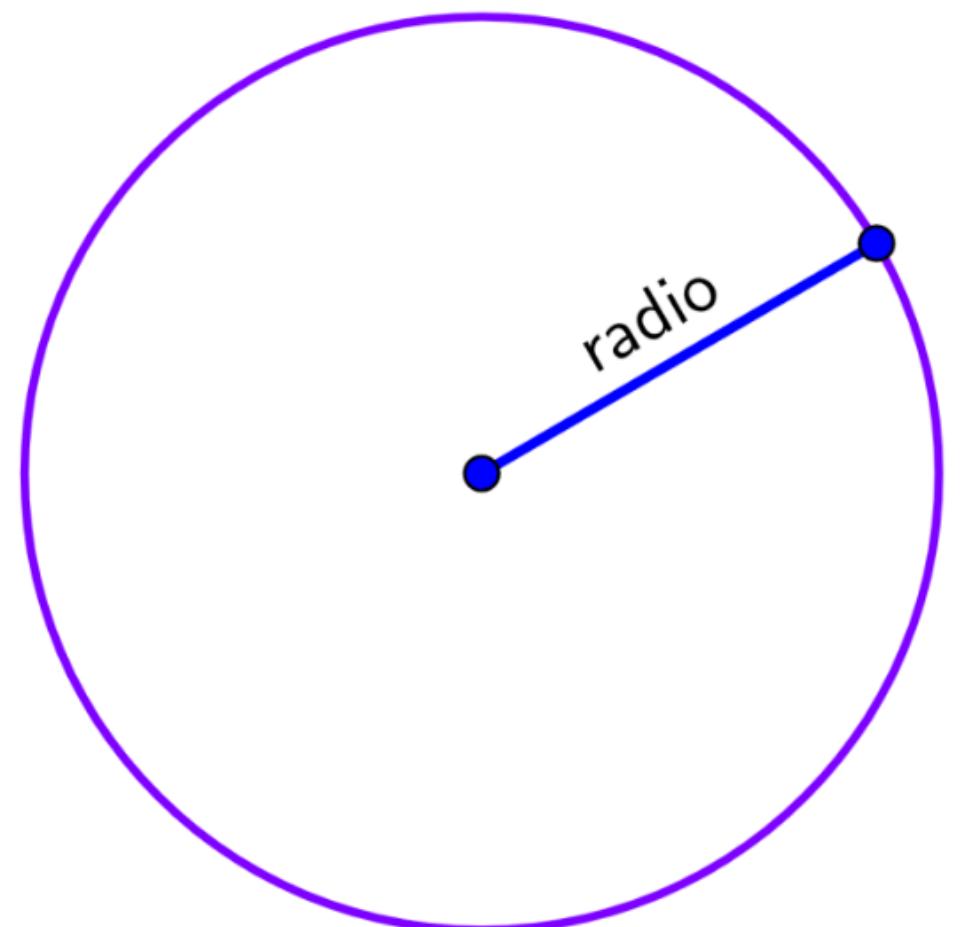
area del círculo

$$\text{Area} = \text{Pi} * (\text{radio}^{**2})$$

Creamos un vínculo entre la palabra
Pi y el valor 3.141



EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141 →

radio = 1.5

area del círculo

Area = Pi * (radio**2)

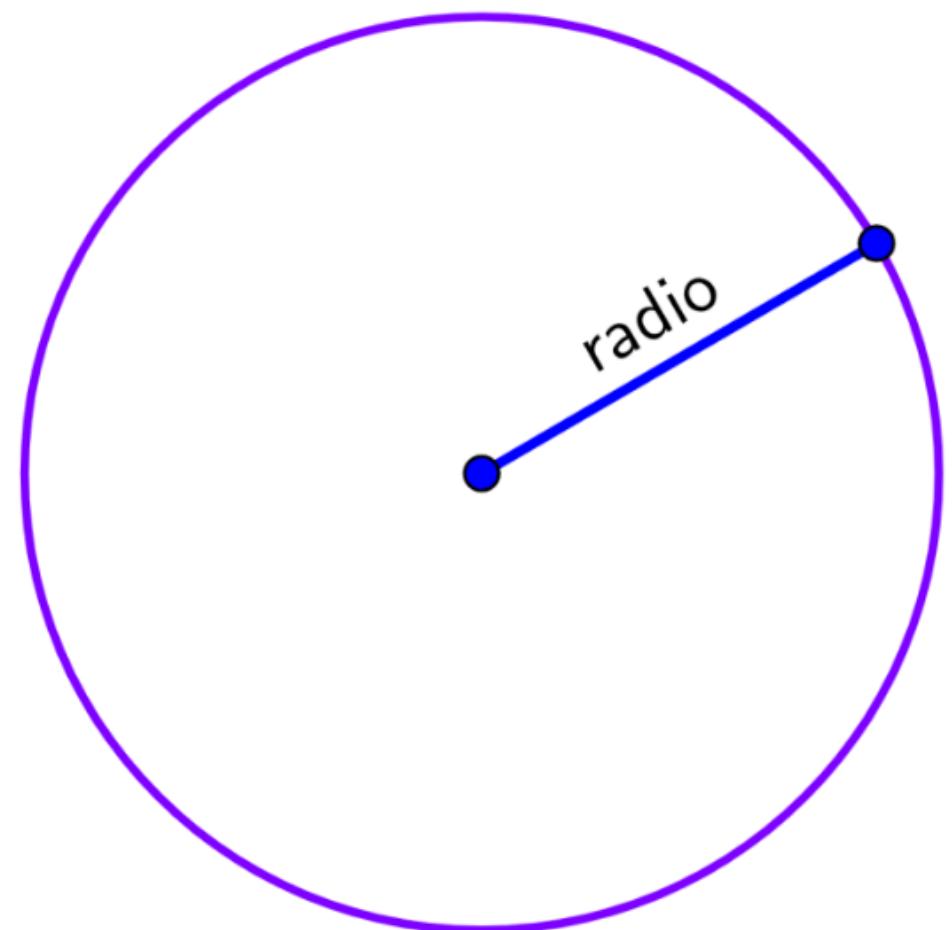
variable

valor

Creamos un vínculo entre la palabra
Pi y el valor 3.141



EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141 →

radio = 1.5

area del círculo

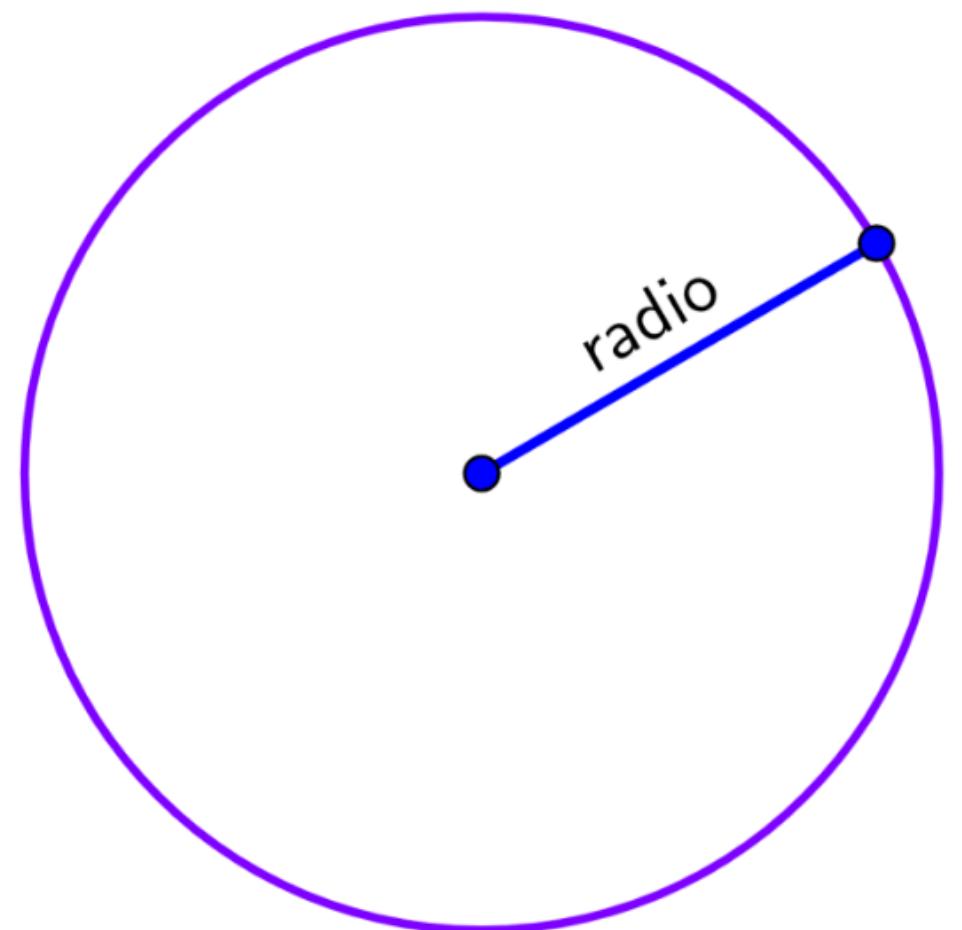
Area = Pi * (radio**2)

Area = 3.14 * (1.5**2)

variable valor

Creamos un vínculo entre la palabra
Pi y el valor 3.141

EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

$\text{Pi} = 3.141$

$\text{radio} = 1.5$

area del círculo

`Area = Pi * (radio**2)`

`Area = 3.14 * (1.5**2)`

¿Por qué usamos variables?

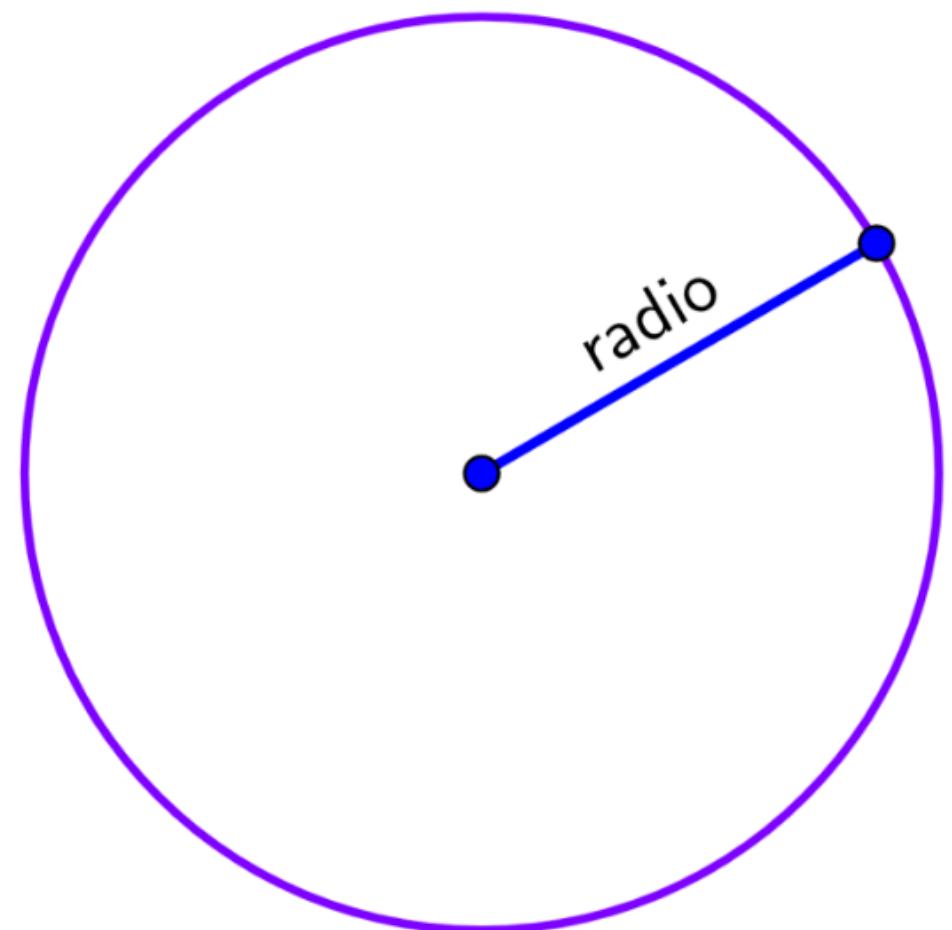
El código queda más claro

Podemos reutilizar valores

Podemos cambiar el código más fácilmente



EJEMPLO: AREA DE UN CÍRCULO



$$A = \pi r^2$$

Pi = 3.141 →

radio = 1.5

area del círculo →

Area = Pi * (radio**2)

Creamos un vínculo entre la palabra
Pi y el valor 3.141

Comentario

Area = 3.14 * (1.5**2)

Pi → Constante

r, Area → Variable

Area = Pi * (radio**2) → Expresión

CONQUER
BLOCKS