

PROGRAMACIÓN I

LABORATORIO I

DOCENTES

José Alberto Rodriguez

JARODRIGUEZ@DOCENTES.FRGP.UTN.EDU.AR

Alejandro Gomez Nieto

ALEJANDRO.GOMEZ@ALUMNOS.FRGP.UTN.EDU.AR

CONTENIDO

01

TIPOS DE DATOS

02

VARIABLES Y CONSTANTES

03

OPERADORES

04

ESTRUCTURA DE SECUENCIA

05

ESTRUCTURA DE DECISIÓN

- SIMPLE

- MULTIPLE

06

ESTRUCTURA DE REPETICIÓN

- CICLO EXACTO

- CICLO INEXACTO

- CICLO COMBINADOS

07

VECTORES Y MATRICES

08

FUNCIONES

09

TIPOS DE PARÁMETROS

10

PROYECTOS DE SOFTWARE

HERRAMIENTAS

Diagrama

- PAPEL Y LAPIZ
- TABLETA
- DRAW.IO
- CANVAS
- PAINT
- ~~NO ES NECESARIO PSEINT~~



SOFTWARE

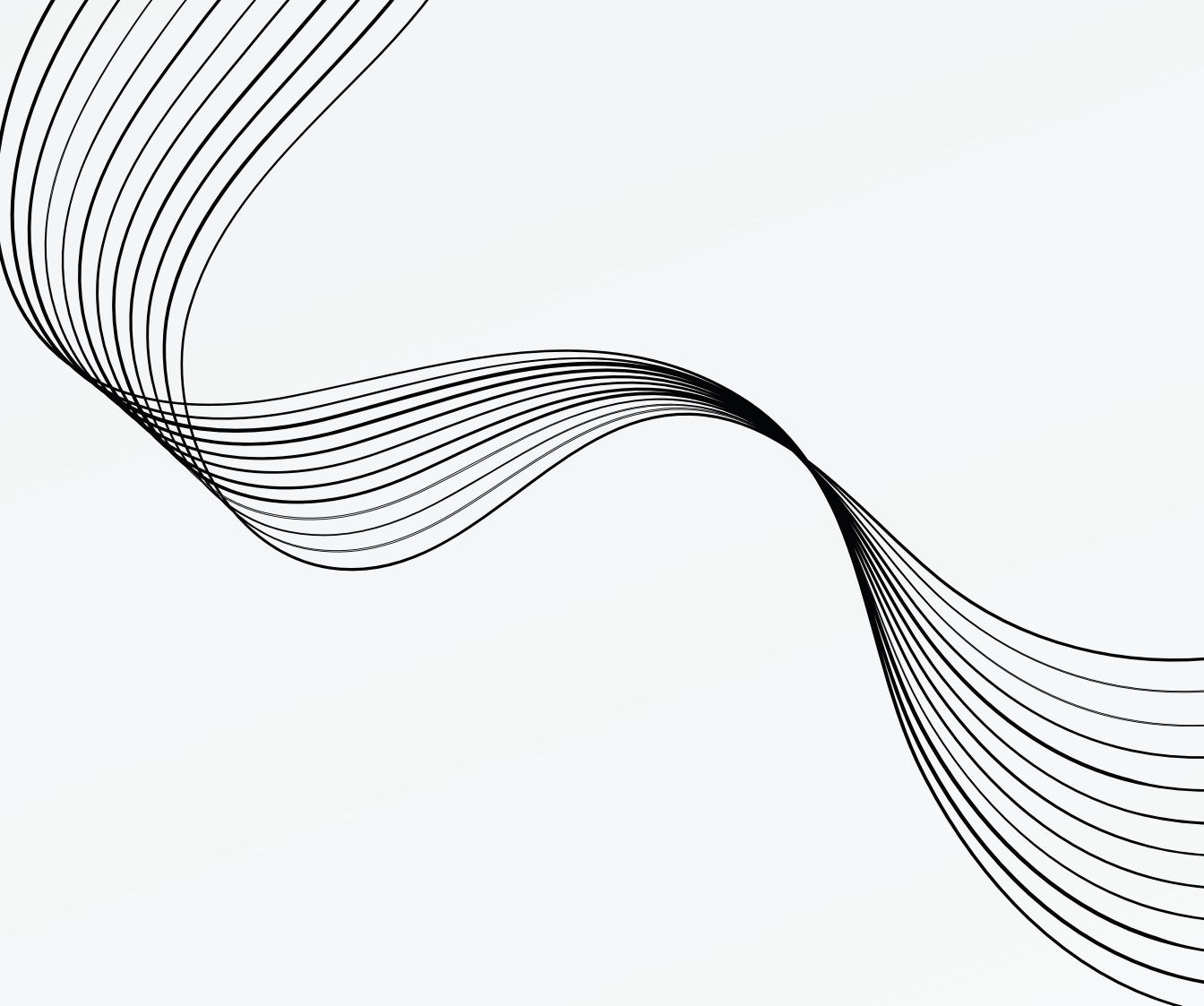
Código

- CODEBLOCKS 20.03 (WINDOWS Y LINUX)
- ZINJAI (WINDOWS)
- DEV C++ (WINDOWS)
- VISUAL STUDIO (WINDOWS)
- VISUAL STUDIO CODE + MINGW (WINDOWS)
- XCODE (MAC)



¿QUÉ ES PROGRAMAR?

*¿Es lo mismo programar
que codificar?*



PROCESO DE DESARROLLO DE UN PROGRAMA

Análisis del problema

Se analizan las características del problema. Se determinan los datos clave de entrada y salida. Se analizan requisitos y restricciones.

Diseño de la solución

Se determina en detalle las estructuras de datos y de programación que se utilizarán. Se hace un esquema general que resuelva claramente el problema.

Codificación

Se eligen las tecnologías más apropiadas para desarrollar la solución y se procede a codificarla.

Compilación

Dependiendo la tecnología, es necesario realizar un proceso de compilación para transformar el código en un programa ejecutable.

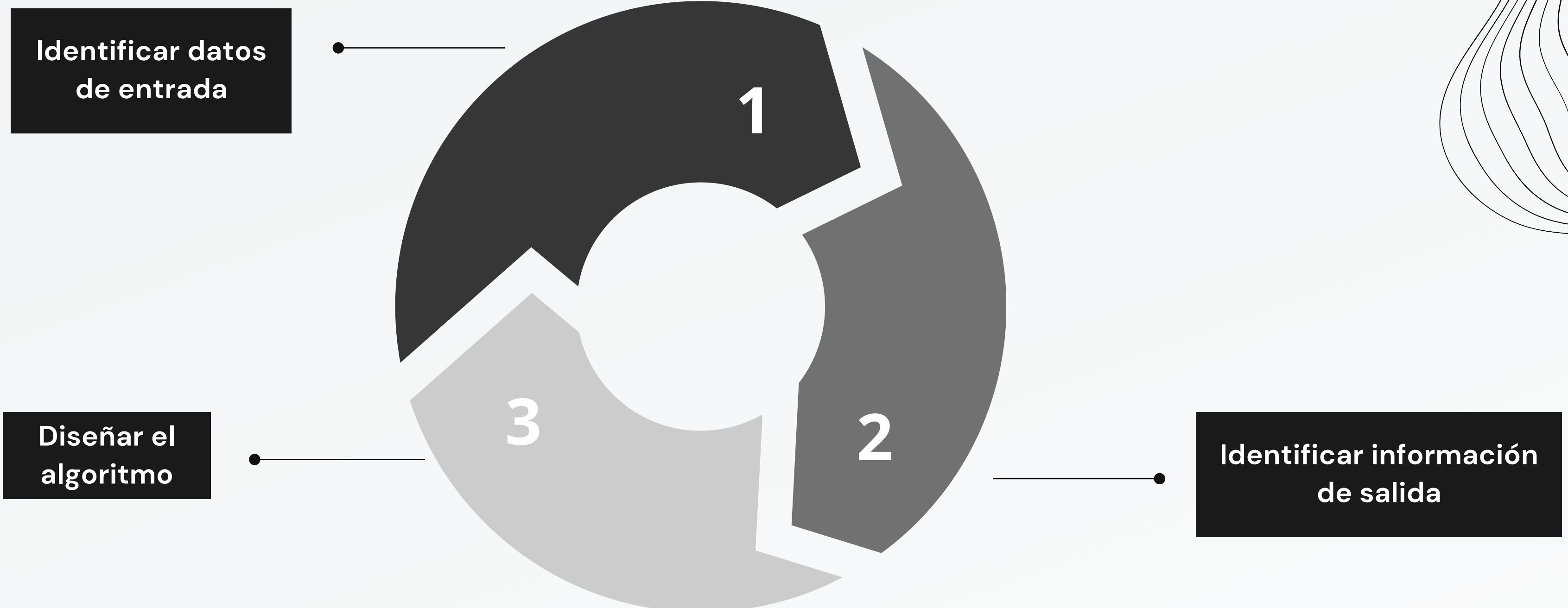
Testing

Se ejecuta el programa creado y se comprueba su correcto funcionamiento.

Instalación y mantenimiento

De ser necesario, se realiza la instalación del programa, su puesta a punto y posterior mantenimiento de solución de bugs y mejoras.

PROCESO DE ANÁLISIS DE UN PROBLEMA



PROCESO DE RESOLUCIÓN DEL PROBLEMA

DATOS DE ENTRADA

PROCESO

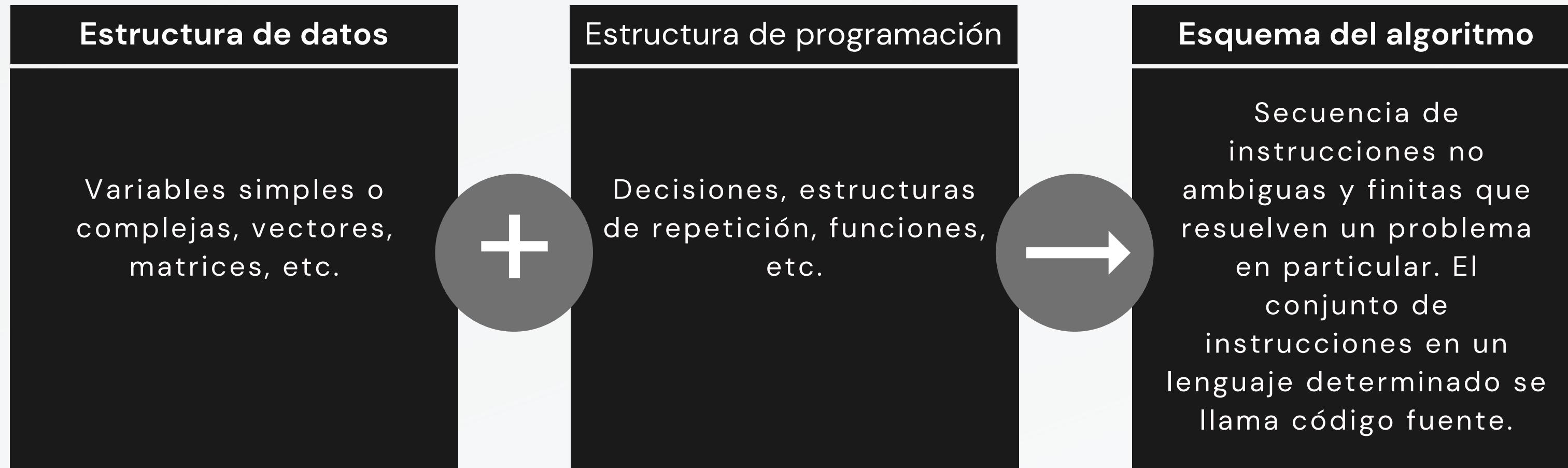
INFORMACIÓN DE SALIDA

Determinar cuántos y cuáles son los datos de entrada de nuestro programa. Ponerles un nombre y determinar su tipo de datos.

El algoritmo ideado debe poder transformar los datos de entrada en la información de salida.
Se apreciará que el algoritmo sea eficiente en su resolución. Resolviendo el problema de la mejor manera y con el menor costo de recursos posibles.

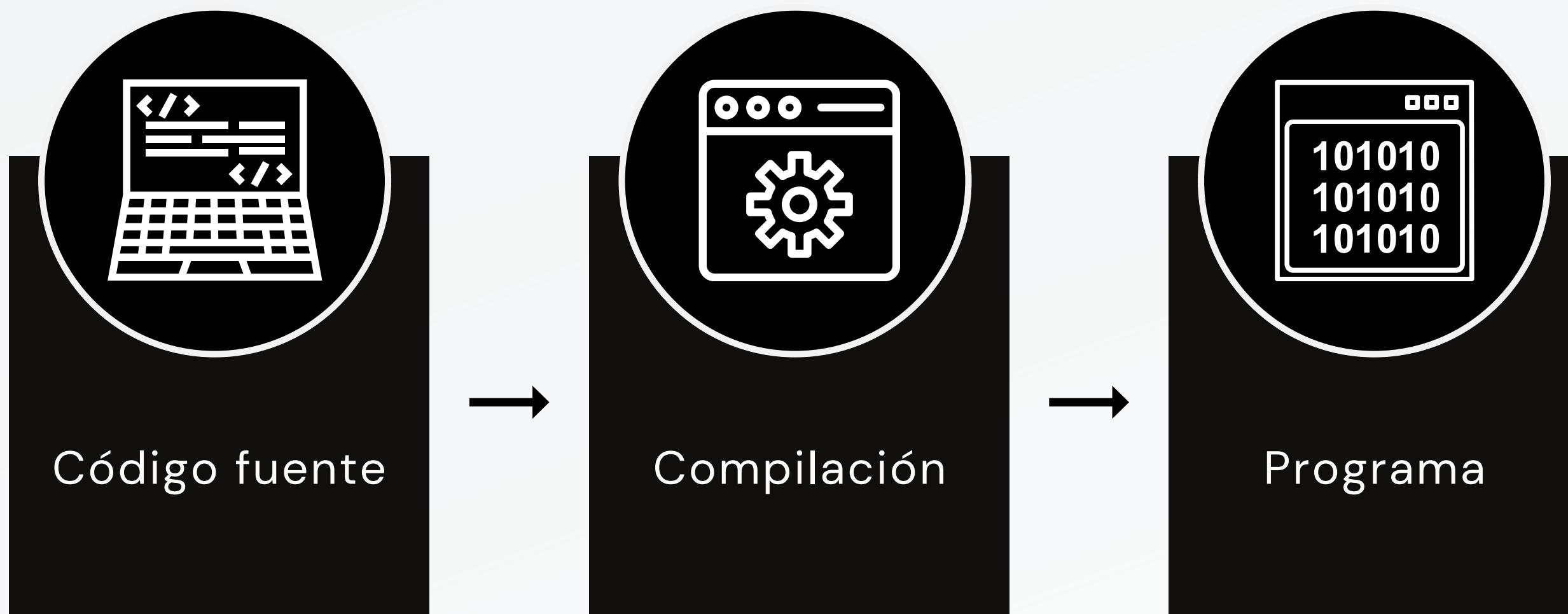
La información de salida debe ser clara, prolífica e informar estrictamente lo necesario.

DISEÑO DE UN ALGORITMO



DEL CÓDIGO AL PROGRAMA

EL PROCESO DE TRANSFORMAR UN CÓDIGO FUENTE EN UN PROGRAMA QUE CONTIENE LAS INSTRUCCIONES QUE SEAN COMPRENSIBLES POR LA COMPUTADORA SE LLAMA COMPILEACIÓN.



CODIFICACIÓN

CADA ELEMENTO QUE UTILICEMOS EN UN **LENGUAJE DE PROGRAMACIÓN** DEBE ESTAR SUJETO A UNA ESTRICTA SINTAXIS. LOS ELEMENTOS QUE EL LENGUAJE ADMITE SON:

- VARIABLES Y CONSTANTES
- OPERADORES
- EXPRESIONES
- PALABRAS RESERVADAS

Como muchos lenguajes, C y C++ son case-sensitive. Esto significa que hace diferencias entre mayúsculas y minúsculas.

VARIABLES

REPRESENTACIÓN SIMBÓLICA DE ESPACIO DE MEMORIA. ES DONDE SE ALMACENAN LOS DATOS EN PROCESAMIENTO.

UNA VARIABLE SE IDENTIFICA CON UN **TIPO DE DATO** Y UN **IDENTIFICADOR DE NOMBRE** (LOS ELIGE EL PROGRAMADOR), Y PERMITE ESCRIBIR UN DATO EN LA MEMORIA O LEER UN DATO DE LA MEMORIA. SE PUEDE MODIFICAR SU VALOR LAS VECES QUE SEA NECESARIO.

```
int edad;  
char caracter;  
float precio;  
bool aprobado;  
string nombre;
```

VARIABLES

EJEMPLO

```
int edad;  
edad = 20;
```

REPRESENTACIÓN SIMBÓLICA DE LA MEMORIA.
AL ESPACIO DE MEMORIA IDENTIFICADO POR **EDAD** SE LE ASIGNÓ EL VALOR 20.



• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
20 edad	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •
• • • •	• • • •	• • • •	• • • •

CONSTANTES

REPRESENTACIÓN SIMBÓLICA DE ESPACIO DE MEMORIA. ES DONDE SE ALMACENAN LOS DATOS EN PROCESAMIENTO.

UNA CONSTANTE SE IDENTIFICA CON LA PALABRA RESERVADA **CONST**, UN TIPO, UN NOMBRE (LO ELIGE EL PROGRAMADOR), Y UN VALOR QUE NO PUEDE SER MODIFICADO DURANTE EL TRANSCURSO DEL PROGRAMA EN EJECUCIÓN.

```
const int EDAD = 20;  
const char LETRA = 'A';  
const float PRECIO = 10.5;
```

OPERADORES

CONJUNTO DE SÍMBOLOS Y PALABRAS RESERVADAS QUE NOS PERMITEN HACER OPERACIONES CON EXPRESIONES.

EXISTEN DIFERENTES CATEGORÍAS DE OPERADORES:



ASIGNACIÓN



ARITMÉTICOS



RELACIONALES



LÓGICOS



DE DIRECCIÓN E INDIRECCIÓN



DE ACCESO A MIEMBROS



PUNTEROS



TERNARIOS

OPERADORES MATEMÁTICOS

NECESARIOS PARA REALIZAR CÁLCULOS MATEMÁTICOS. LOS PARÉNTESIS TIENEN EL MISMO EFECTO QUE EN LA MATEMÁTICA EN LAS EXPRESIONES ALGEBRAICAS. SIN EMBARGO, EN PROGRAMACIÓN NO SE UTILIZAN CORCHETES NI LLAVES PARA LA SPARACIÓN DE TÉRMINOS.

Operador	Operación
+	Suma
-	Resta
*	Producto
/	División
%	Resto

$((2+3)*5) + 10$
$2+3*5+10$
$5 / 2$
$5 . 0 / 2$
$5 \% 2$

OPERADORES MATEMÁTICOS

NECESARIOS PARA REALIZAR CÁLCULOS MATEMÁTICOS. LOS PARÉNTESIS TIENEN EL MISMO EFECTO QUE EN LA MATEMÁTICA EN LAS EXPRESIONES ALGEBRAICAS. SIN EMBARGO, EN PROGRAMACIÓN NO SE UTILIZAN CORCHETES NI LLAVES PARA LA SPARACIÓN DE TÉRMINOS.

Operador	Operación
+	Suma
-	Resta
*	Producto
/	División
%	Resto

$((2+3)*5)+10 \longrightarrow 35$
$2+3*5+10 \longrightarrow 27$
$5/2 \longrightarrow 2$
$5.0/2 \longrightarrow 2.5$
$5\%2 \longrightarrow 1$

DIVISIÓN REAL VS DIVISIÓN ENTERA

UNA DIVISIÓN PUEDE RESOLVERSE DE DOS MANERAS. OBTENIENDO UN RESULTADO CON EXPRESIÓN DECIMAL O RESULTADOS ENTEROS. VEAMOS ESTOS EJEMPLOS:



TENEMOS SEIS KILOS DE HELADO PARA
REPARTIR ENTRE OCHO PERSONAS.
¿CUÁNTOS KILOS LES CORRESPONDE A
CADA UNO?

TENEMOS DIECISIETE CHUPETINES PARA
REPARTIR ENTRE CINCO PERSONAS.
¿CUÁNTAS CHUPETINES LES
CORRESPONDE A CADA UNO?

DIVISIÓN REAL VS DIVISIÓN ENTERA



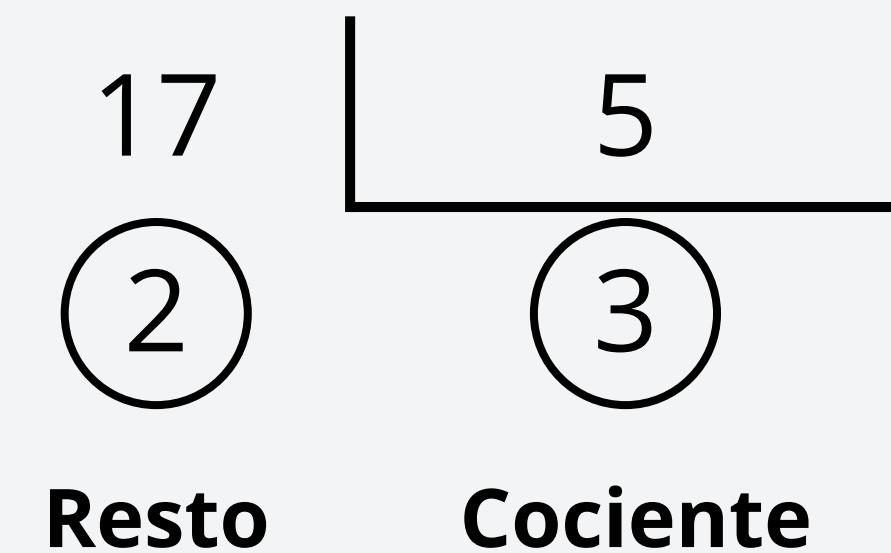
TENEMOS SEIS KILOS DE HELADO PARA
REPARTIR ENTRE OCHO PERSONAS.
¿CUÁNTOS KILOS LES CORRESPONDE A
CADA UNO?



CORRESPONDE 0.75 KILOS POR PERSONA



TENEMOS DIECISIETE CHUPETINES PARA
REPARTIR ENTRE CINCO PERSONAS.
¿CUÁNTAS CHUPETINES LES
CORRESPONDE A CADA UNO?



CORRESPONDE 3 CHUPETINES POR PERSONA Y SOBRAN 2 CHUPETINES

OPERADORES RELACIONALES

Son necesarios para decisiones y ciclos. Nos permiten establecer proposiciones lógicas. El resultado de una proposición lógica puede ser **verdadero** o **falso**.

Ejemplos

Si me pregunto **5 > b**, el resultado va a depender en función del valor de la variable b.

- Si b es igual a 6, el resultado es **falso**.
- Si b es igual a 1, el resultado es **verdadero**.

Operador	Significado
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que
==	Igual que
!=	Distinto que

OPERADORES DE ASIGNACIÓN

NOS PERMITE ASIGNAR LA EXPRESIÓN QUE SE ENCUENTRA A LA DERECHA DEL OPERADOR EN LA VARIABLE QUE SE ENCUENTRA A LA IZQUIERDA.



```
edad = 20;  
a = b;  
nombre = "Pedro";  
precio = cant * pu;
```



```
20 = edad;  
20 = 30;
```

SIN SENTIDO

```
edad = edad;
```

EXPRESIONES

CONJUNTO DE VARIABLES, CONSTANTES, NÚMEROS Y OPERADORES ORDENADOS DE ACUERDO A LAS REGLAS SINTÁCTICAS ESTABLECIDAS EN EL LENGUAJE DE PROGRAMACIÓN.

TIENEN COMO OBJETIVO LA CONSTRUCCIÓN DE INSTRUCCIONES PARA LA RESOLUCIÓN DEL PROBLEMA (O DE PARTE DEL PROBLEMA) PLANTEADO.

10

50 + 10

aux - 20

'B'

PALABRAS RESERVADAS

Palabras que el lenguaje utiliza para identificar tipos de datos, estructuras de programación, etc. Tienen un significado especial para el lenguaje y no pueden ser utilizados como **identificadores de nombre**.

Tipos de datos	bool, int, float, char, short, void, long, double
Elementos de programación	if, else, switch, default, break, for, while, do, return, auto, struct, class, static, virtual
Operadores	new, delete, sizeof

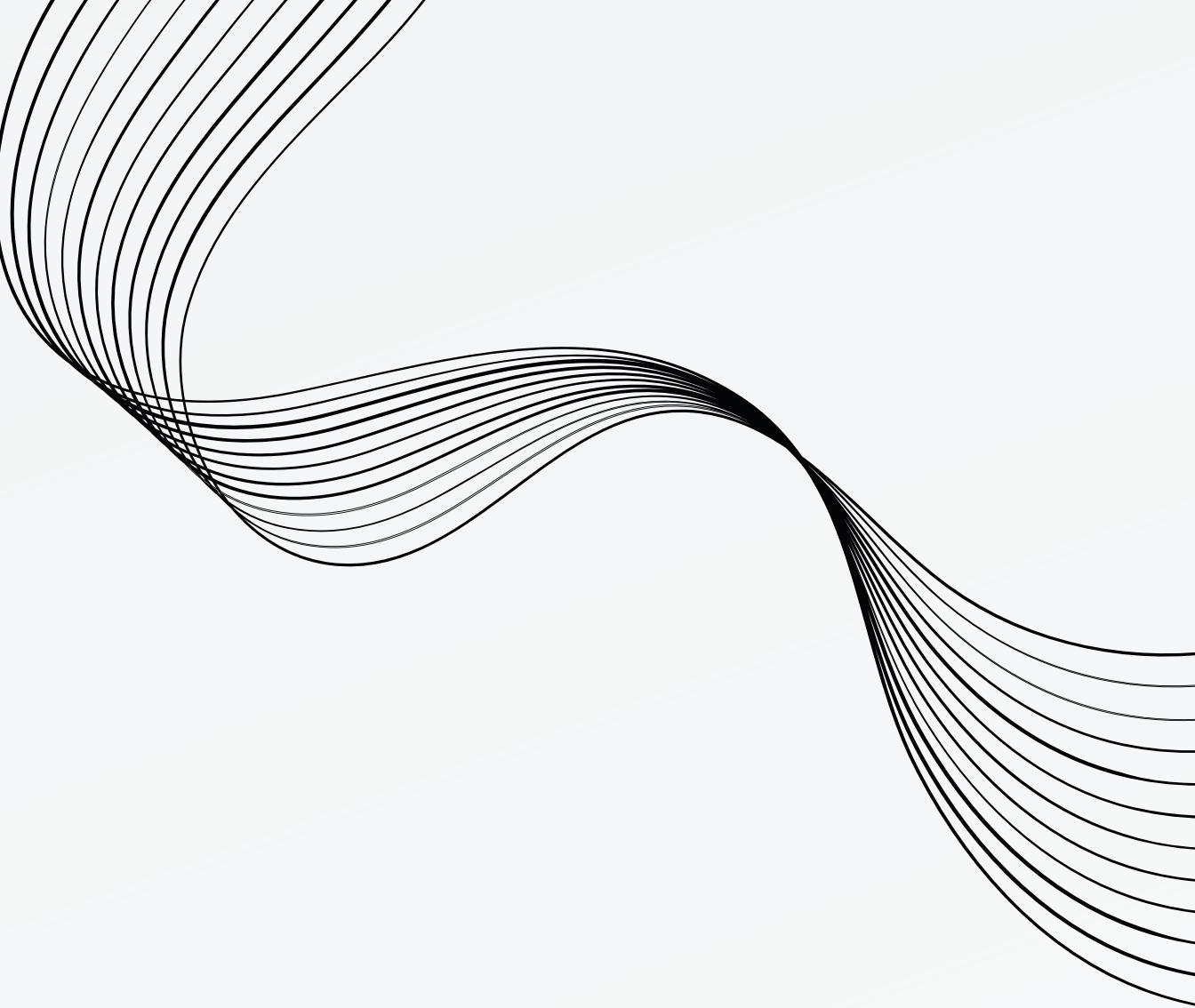
EJERCICIO

HACER UN PROGRAMA QUE PERMITA INGRESAR DOS NÚMEROS ENTEROS POR TECLADO. LUEGO CALCULAR E INFORMAR LA SUMA DE ELLOS.



RESOLUCIÓN DEL PROBLEMA

Diagrama



RESOLUCIÓN: INSTRUCCIONES

O1

DECLARAR LAS VARIABLES
NO ES OBLIGATORIO

```
int num1, num2, resultado
```

O2

INGRESAR EL PRIMER NÚMERO
POR TECLADO

```
num1
```

O3

INGRESAR EL SEGUNDO
NÚMERO POR TECLADO

```
num2
```

O4

REALIZAR LA SUMA

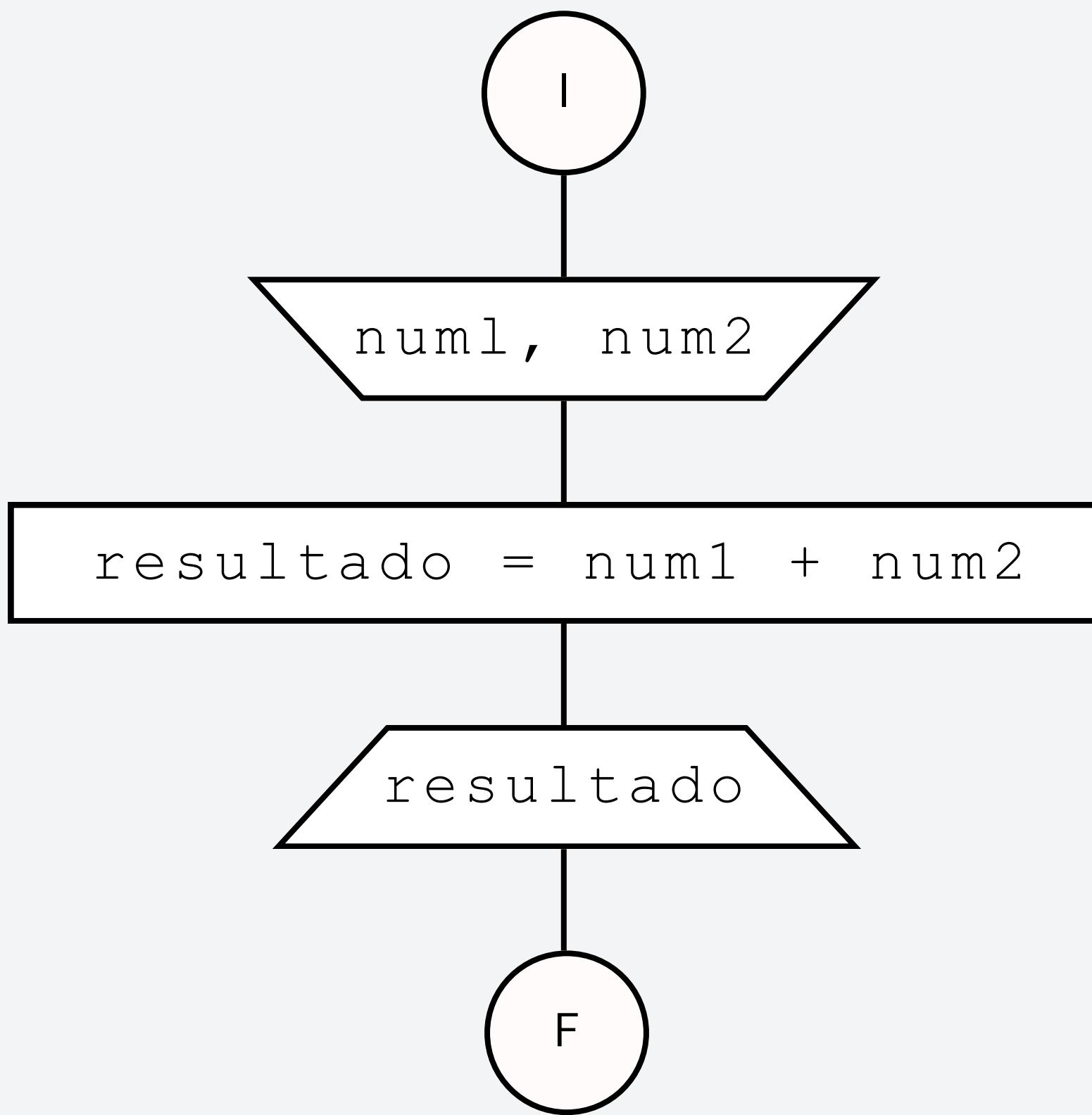
```
resultado = num1 + num2
```

O5

MOSTRAR POR PANTALLA EL
RESULTADO

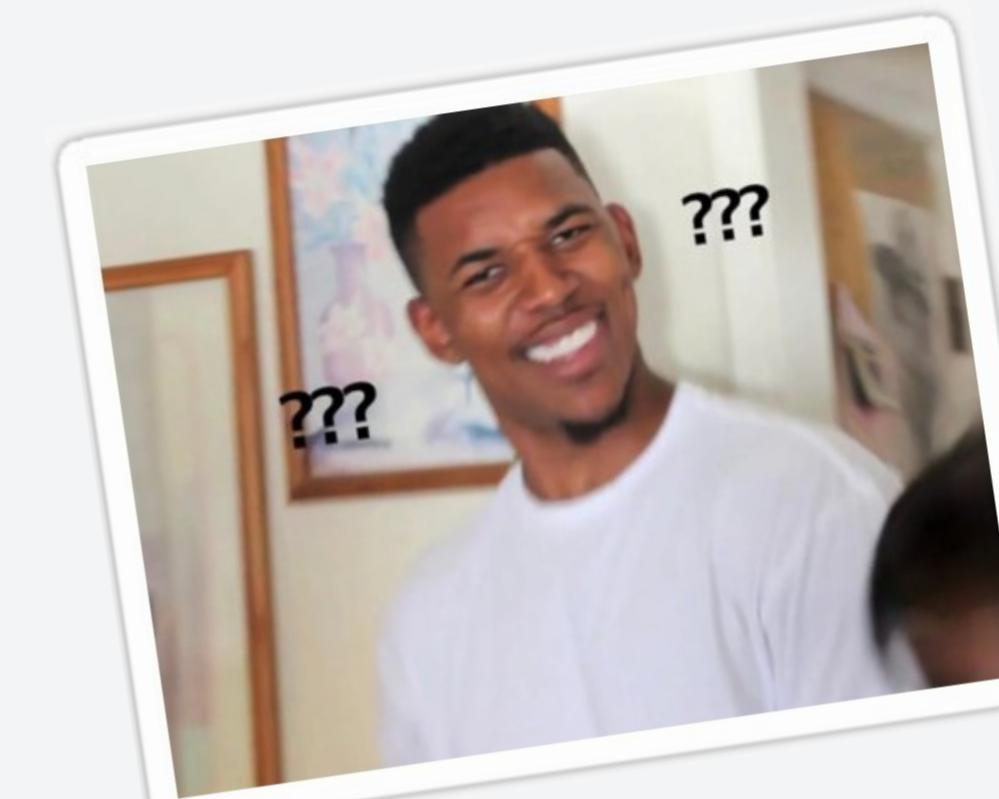
```
resultado
```

RESOLUCIÓN: DIAGRAMA



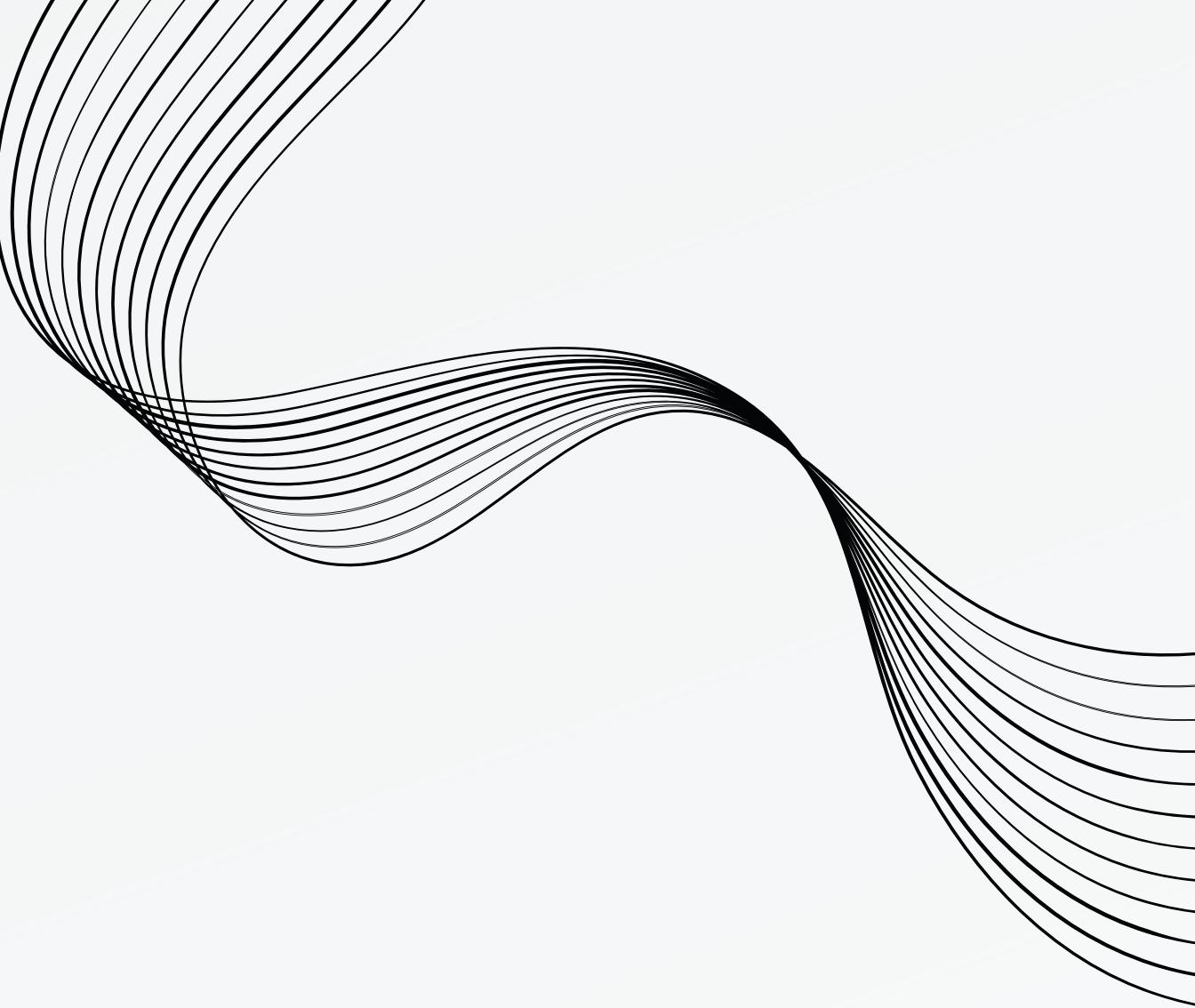
¿QUÉ ES TODO ESO?

NO SE PREOCUPEN. LO
VAMOS A EXPLICAR
PASO A PASO MÁS
ADELANTE.



RESOLUCIÓN DEL PROBLEMA

Código



RESOLUCIÓN: INSTRUCCIONES (CÓDIGO)

O1	DECLARAR LAS VARIABLES NECESARIAS	int num1, num2, resultado;
O2	INGRESAR EL PRIMER NÚMERO POR TECLADO	cin >> num1;
O3	INGRESAR EL SEGUNDO NÚMERO POR TECLADO	cin >> num2;
O4	REALIZAR LA SUMA	resultado = num1 + num2;
O5	MOSTRAR POR PANTALLA EL RESULTADO	cout << resultado;

RESOLUCIÓN: CÓDIGO FUENTE

```
#include <iostream>
using namespace std;

int main(){
    int num1, num2, resultado;
    cin >> num1;
    cin >> num2;
    resultado = num1 + num2;
    cout << resultado;
    return 0;
}
```

¿QUÉ ES TODO ESO?

NO SE PREOCUPEN. LO
VAMOS A CODIFICAR EN UN
IDE PASO A PASO.



