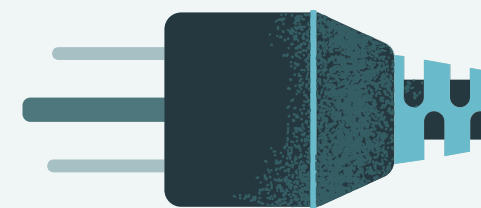
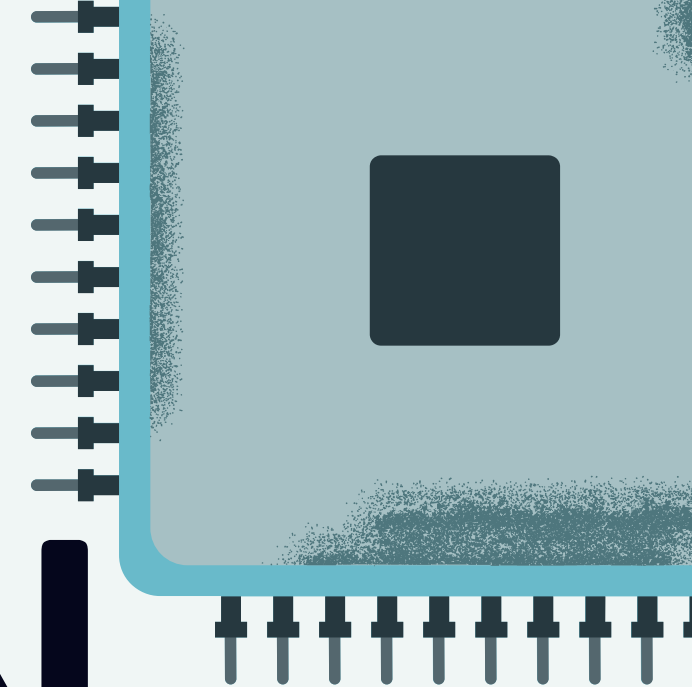
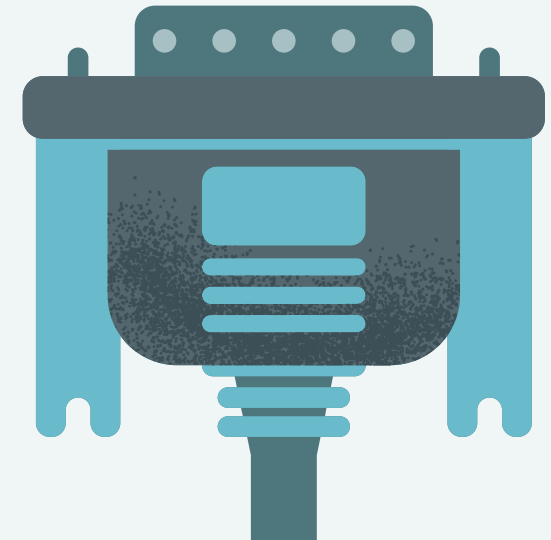
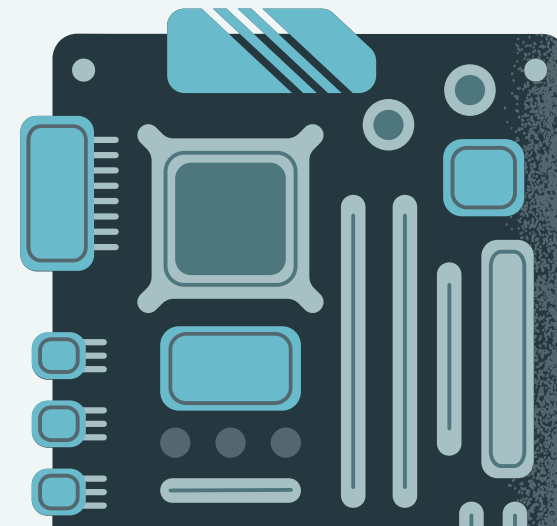
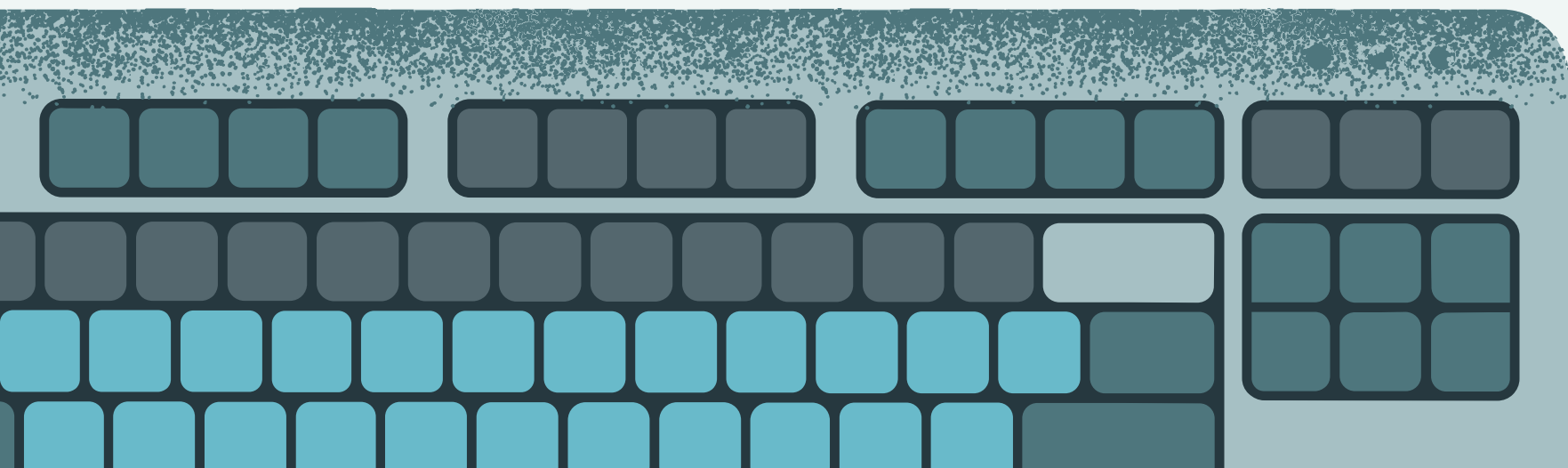
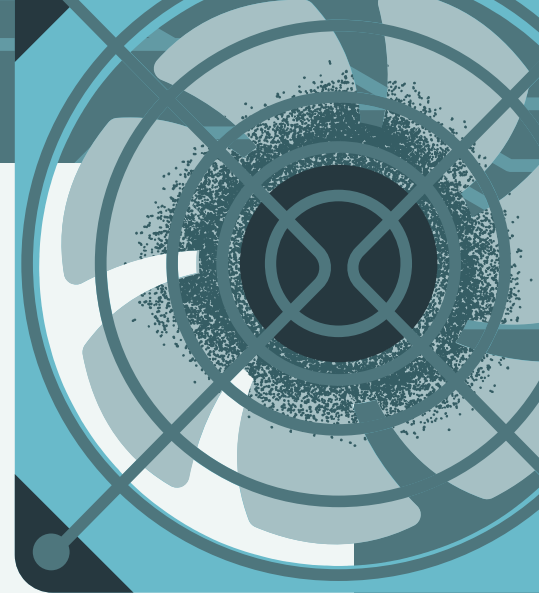
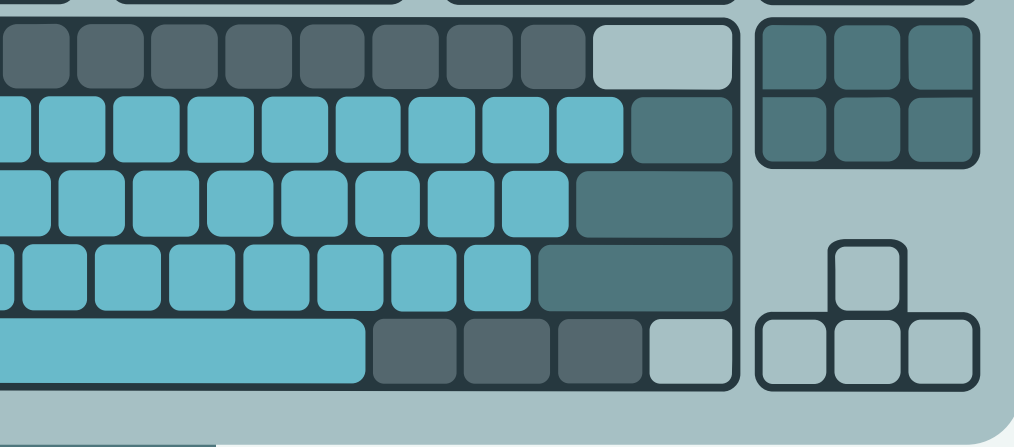


<!-- Tema 2-->

PROGRAMACIÓN BÁSICA

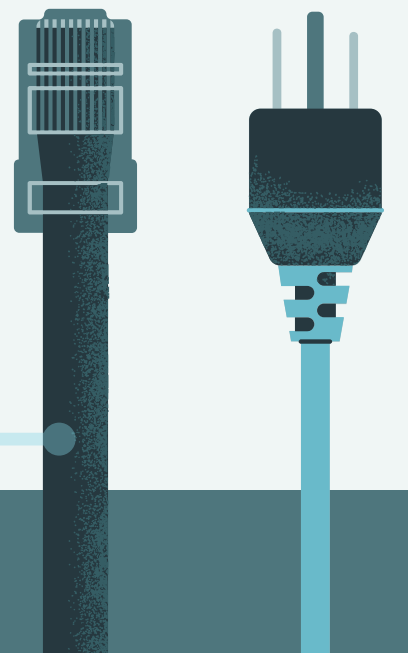
<:: Diseño de algoritmos>



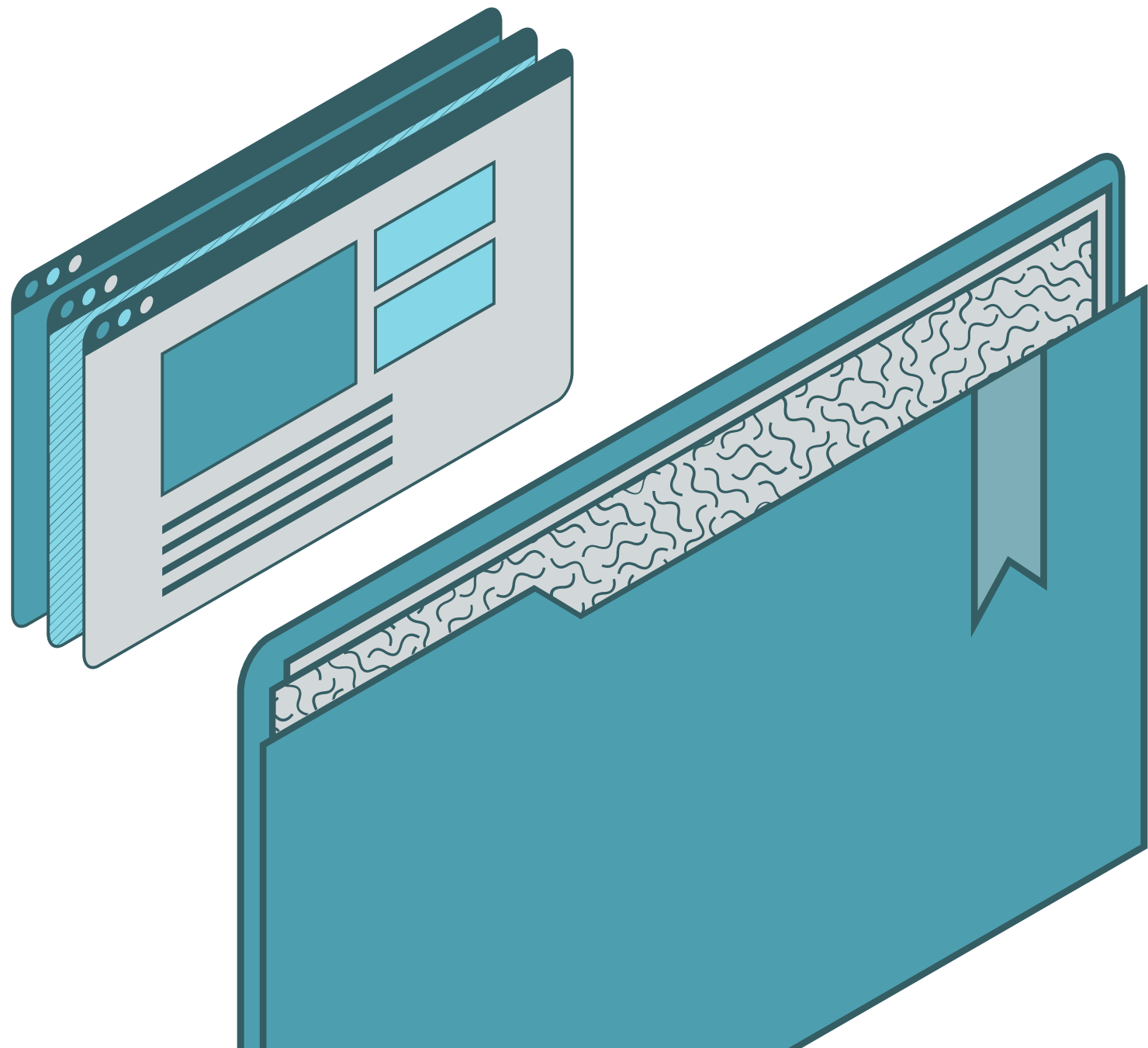


OBJETIVO

El alumno construirá algoritmos para
resolver problemas.



ÍNDICE



2.1

Definición, planteamiento y
modelado del problema.

2.2

Desarrollo de algoritmos.

2.3

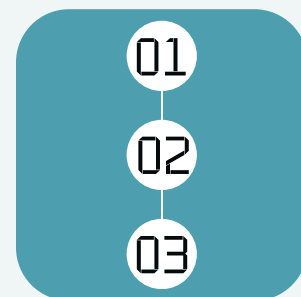
Verificación y depuración de
algoritmos.

CICLO DE VIDA DE UN PROGRAMA



01

Análisis del
problema



02

Elaborar el
algoritmo



03

Codificación del
algoritmo



04

Depuración del
código



05

Mantenimiento del
programa

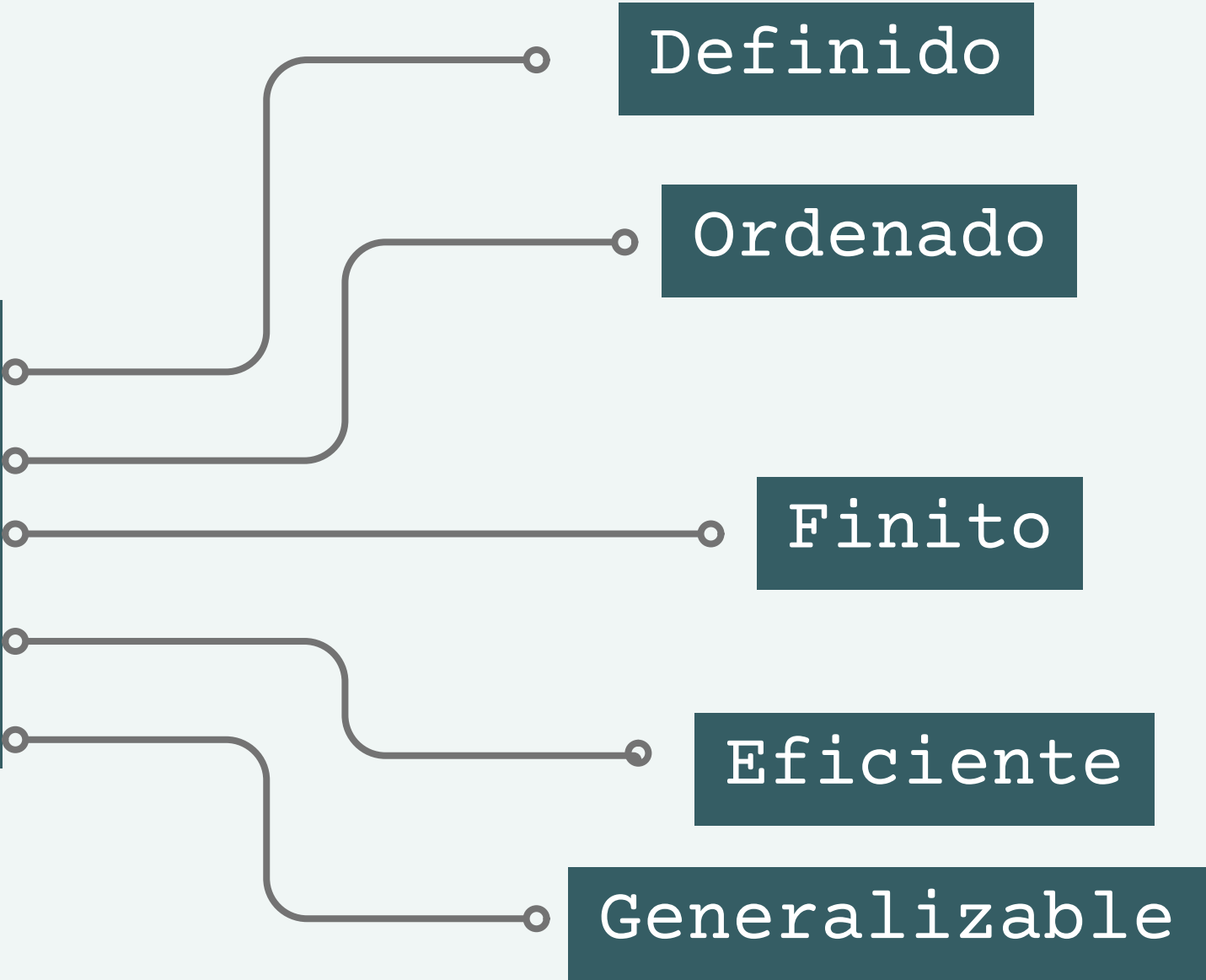
ANÁLISIS DEL PROBLEMA

Proceso de entender claramente qué se quiere resolver.
Incluye identificar entradas, procesos y salidas.



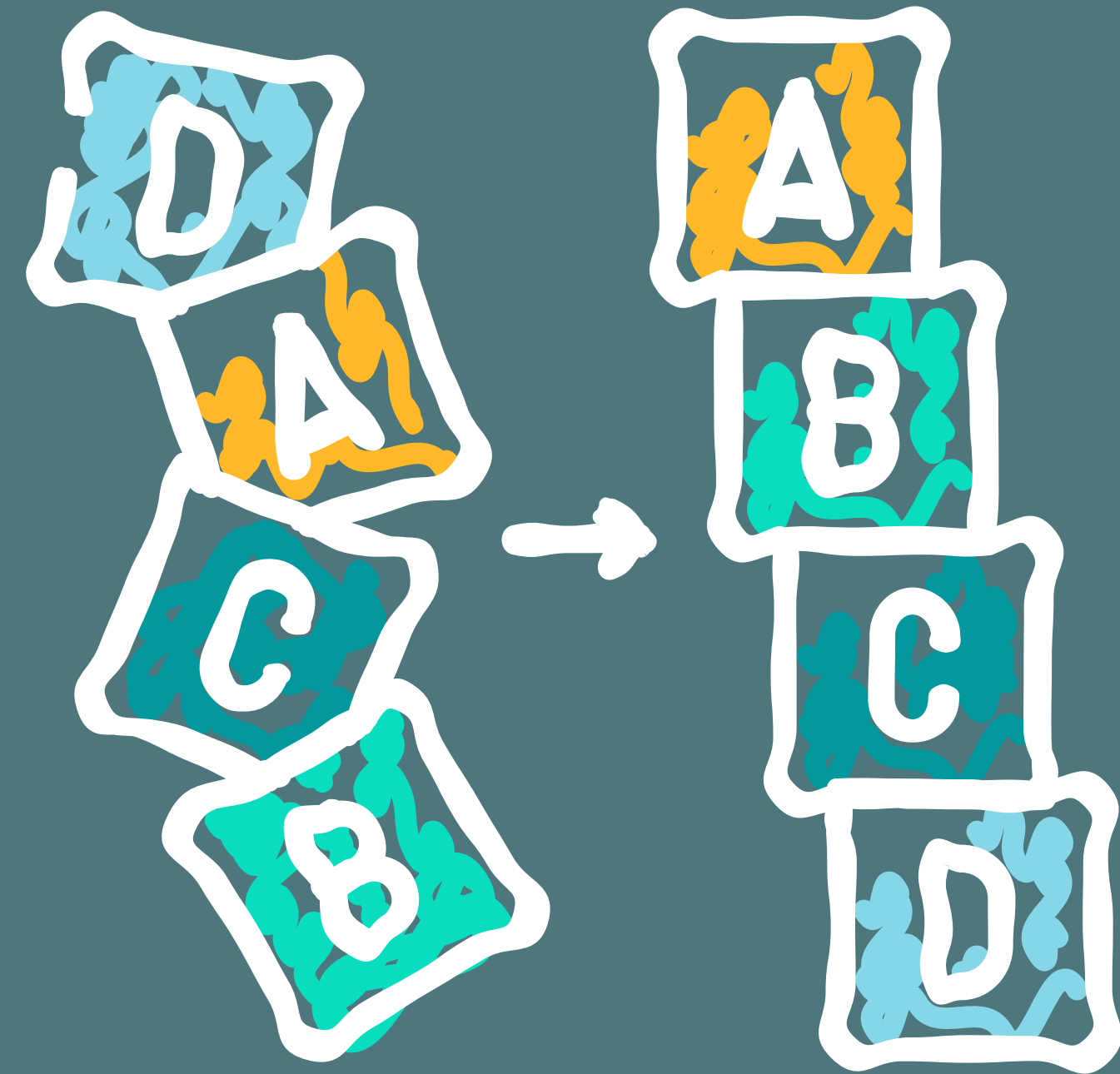
ELABORAR EL ALGORITMO

Un **algoritmo** es un conjunto finito de pasos que permiten resolver un problema o realizar una tarea.

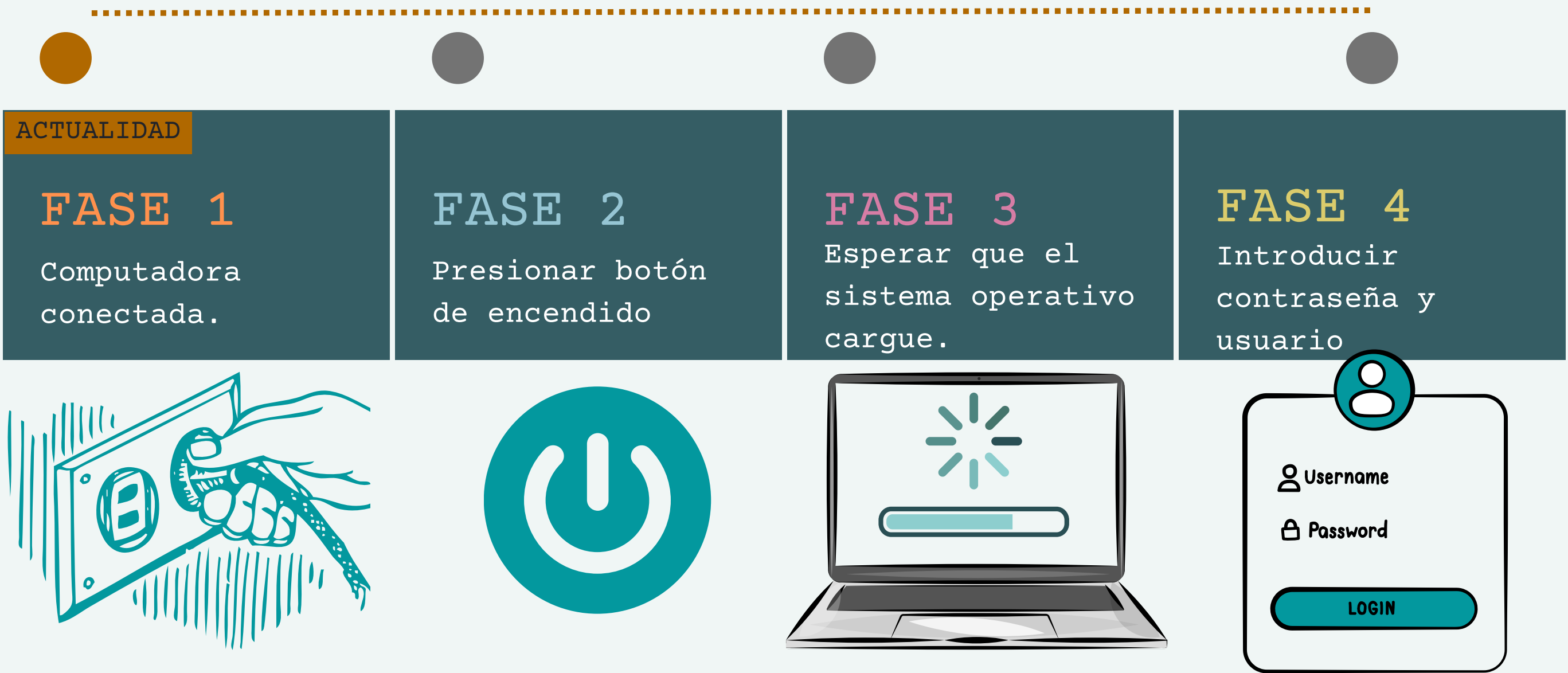


ELABORAR EL ALGORITMO

Debe pensarse detallada y ordenadamente todos los pasos que realizará el programa. Puede ser texto (pseudocódigo) o grafico (diagrama de flujo)



EJEMPLO: ENCENDER COMPUTADORA



CODIFICACIÓN DEL ALGORITMO

En esta etapa se tiene que traducir el algoritmo a un lenguaje de programación específico, las acciones definidas las vamos a convertir en instrucciones.



DEPURACIÓN DEL CÓDIGO

Después de codificar el programa, generalmente quedan detalles a corregir, por lo tanto, el siguiente paso es la depuración del programa.



Pulir el programa para que todo funcione como deseamos. Necesario probar el programa, para asegurarnos que funcione correctamente



MANTENIMIENTO DEL PROGRAMA

Se realiza después de que el programa ya ha tenido vida útil y debido a las nuevas necesidades de los usuarios, es necesario hacer modificaciones al programa

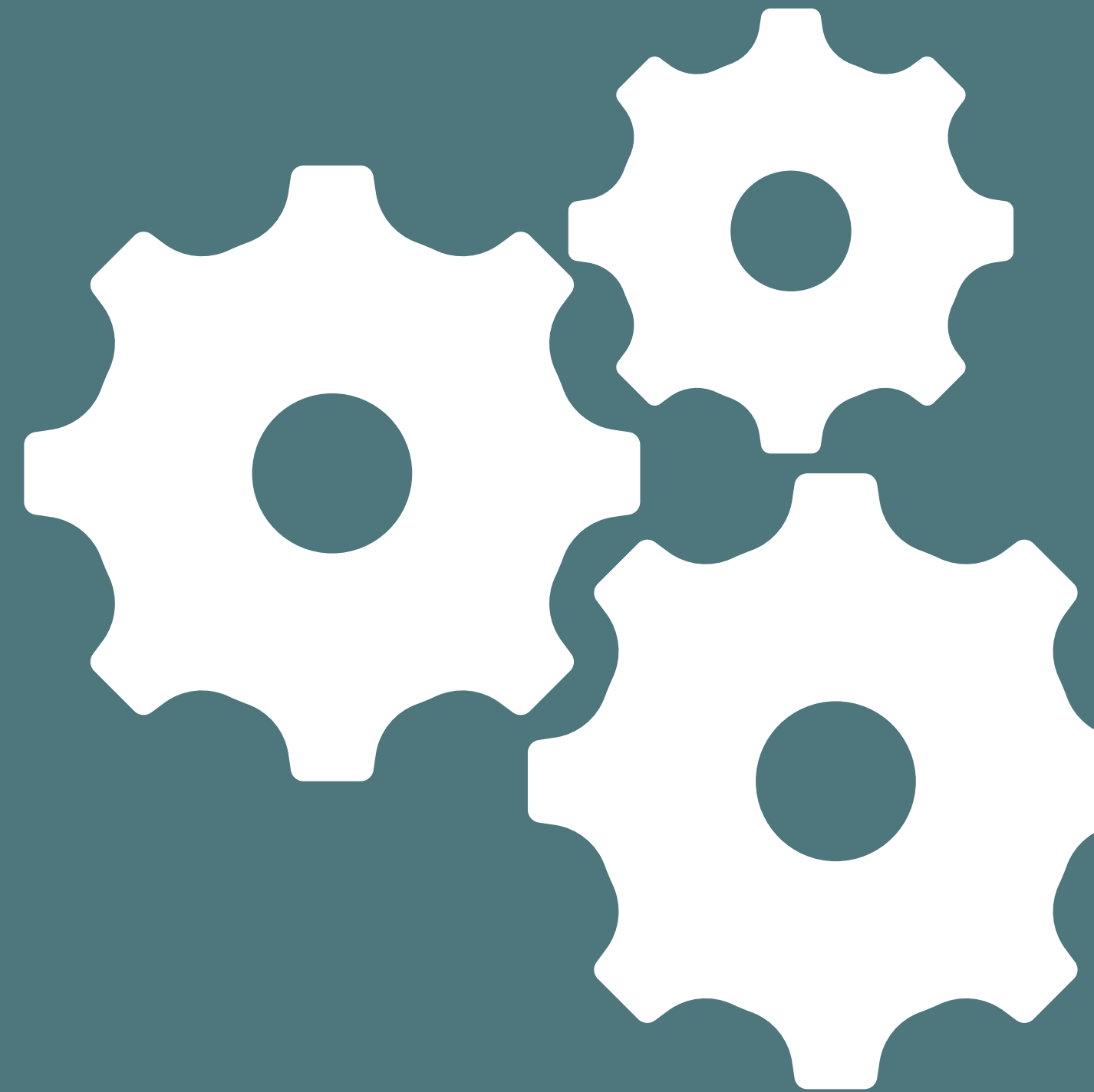


DIAGRAMA DE FLUJO



¿QUÉ ES?

- Es la representación grafica de las secuencias lógicas que se realizan para la resolución de un problema.

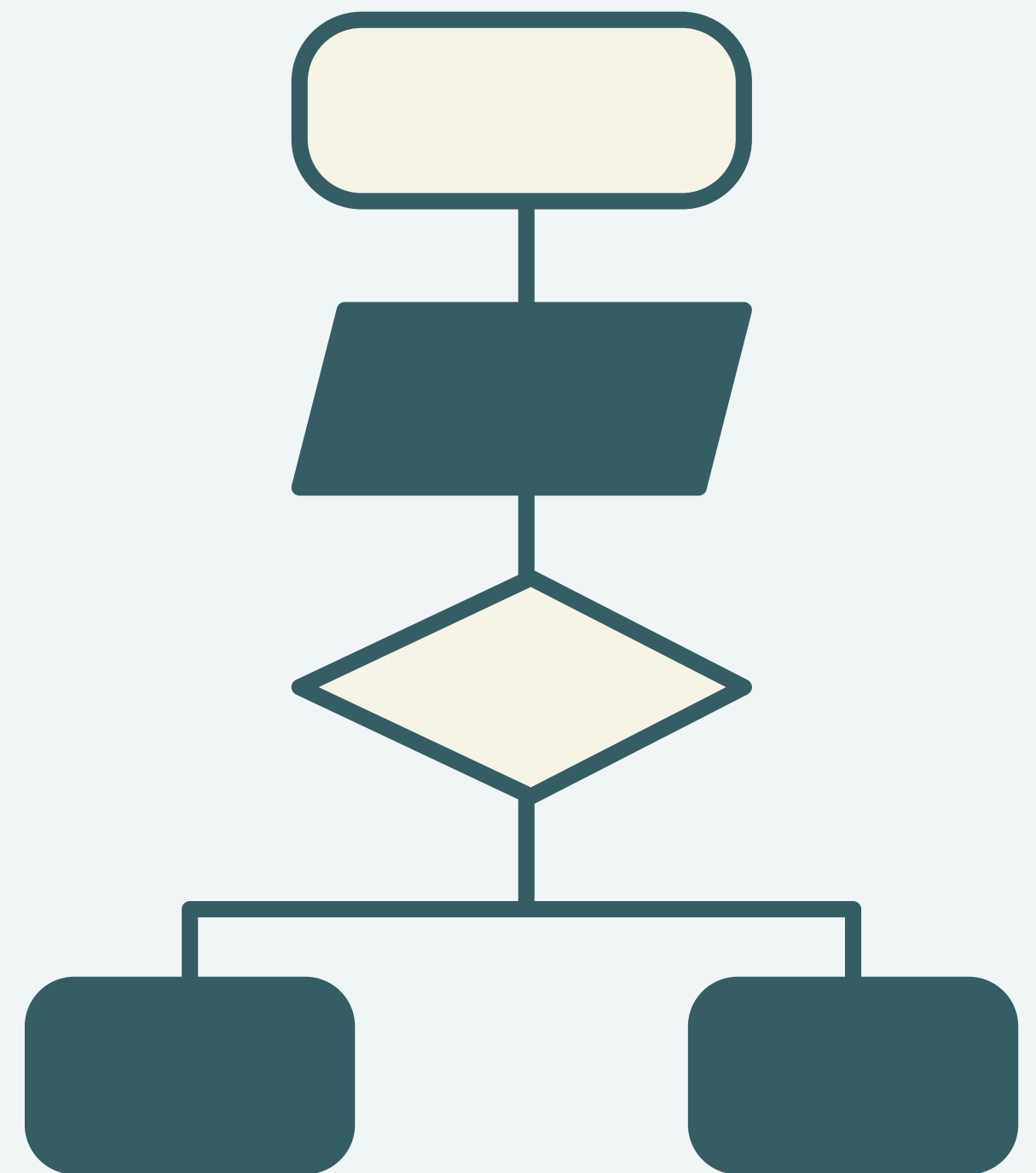


¿PARA QUÉ SIRVE?

- Permite visualizar de forma clara la secuencia de acciones antes de programar o ejecutar el proceso.



Describe de manera grafica la estructura de un algoritmo.



RECOMENDACIONES PARA ELABORAR UN DIAGRAMA DE FLUJO



COMIENZO DEL DIAGRAMA EN LA PARTE SUPERIOR



FIN DEL DIAGRAMA PARTE INFERIOR



EL FLUJO DE LAS SECUENCIAS DEBE IR DE ARRIBA - ABAJO



EVITAR CRUCES DE LINEAS DE FLUJO



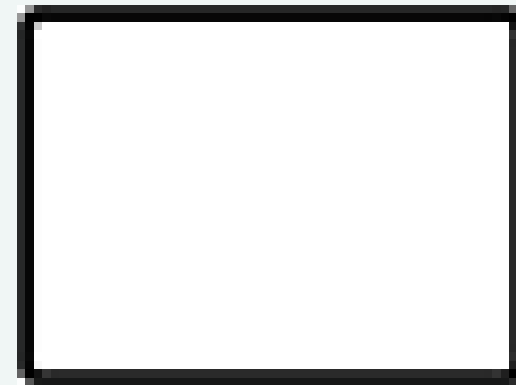
SIMBOLO DE INICIO Y FIN DEBEN DE ESTAR SOLO UNA VEZ

ELEMENTOS DIAGRAMA DE FLUJO



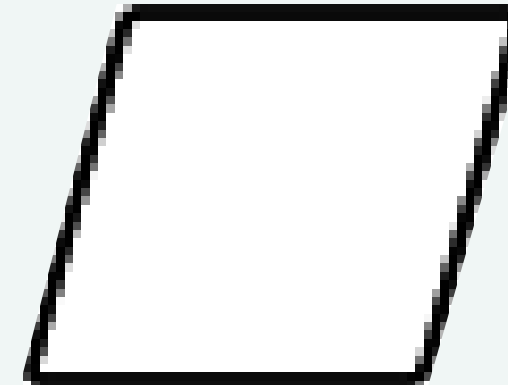
INICIO / FIN

Delimita inicio y fin de un diagrama



PROCESO

Cualquier tipo de operación (asignación, operaciones aritméticas, etc)



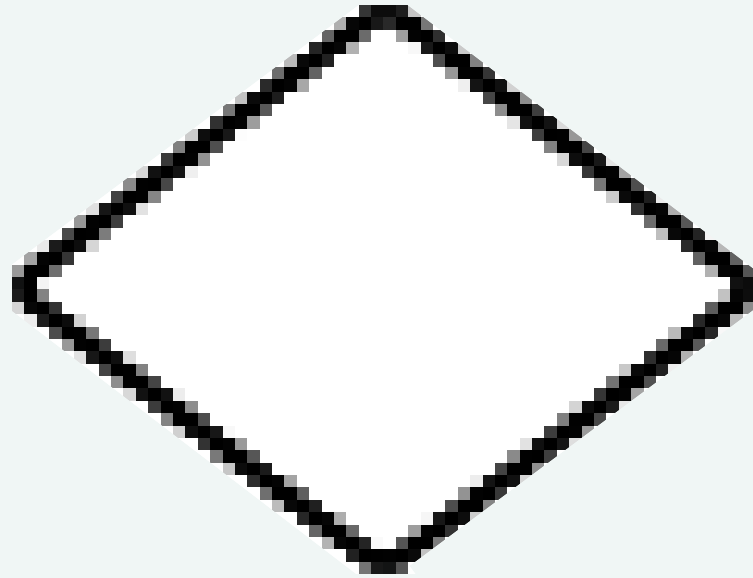
ENTRADA DE DATOS

Introducción de datos



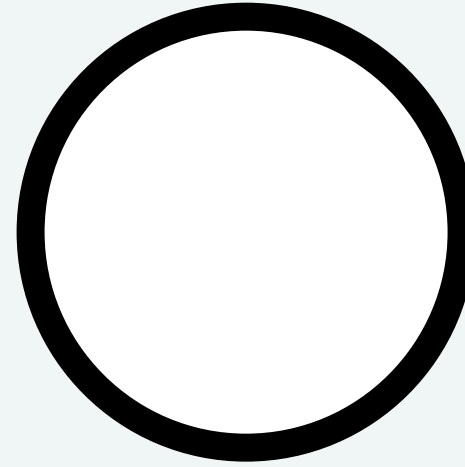
SALIDA DE DATOS

Impresión de un resultado.



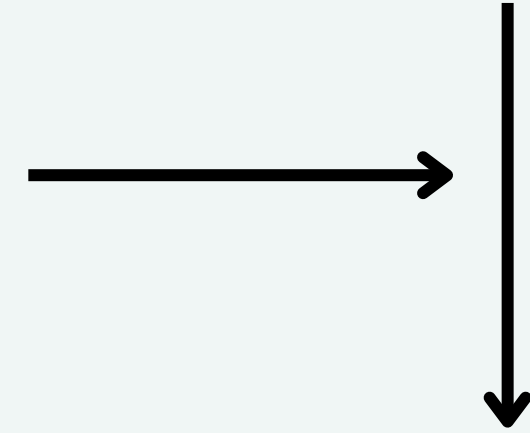
ELEMENTO DE DESICION

Operaciones lógicas o de comparación entre datos. Determina que camino seguir, tiene dos salidas, respuesta si o respuesta no.



CONECTOR

Indica que el flujo del proceso continua en otra parte del diagrama



FLUJO DE DATOS

Indican la dirección y secuencia del proceso, conectando un paso con el siguiente.

INICIO

$Z = A + B$

X

X

INICIO

ASIGNAR $Z = A + B$

LEER X

MOSTRAR X

FIN

FIN

X, Y

LEER X, Y

"TEXTO"
X, Y

MOSTRAR "TEXTO", X, Y

OPERADORES

Paréntesis

()

Potencia

**

Multiplicación

*

División

/

Modulo (residuo)

%

Suma

+

Resta

-

Operador de asignación

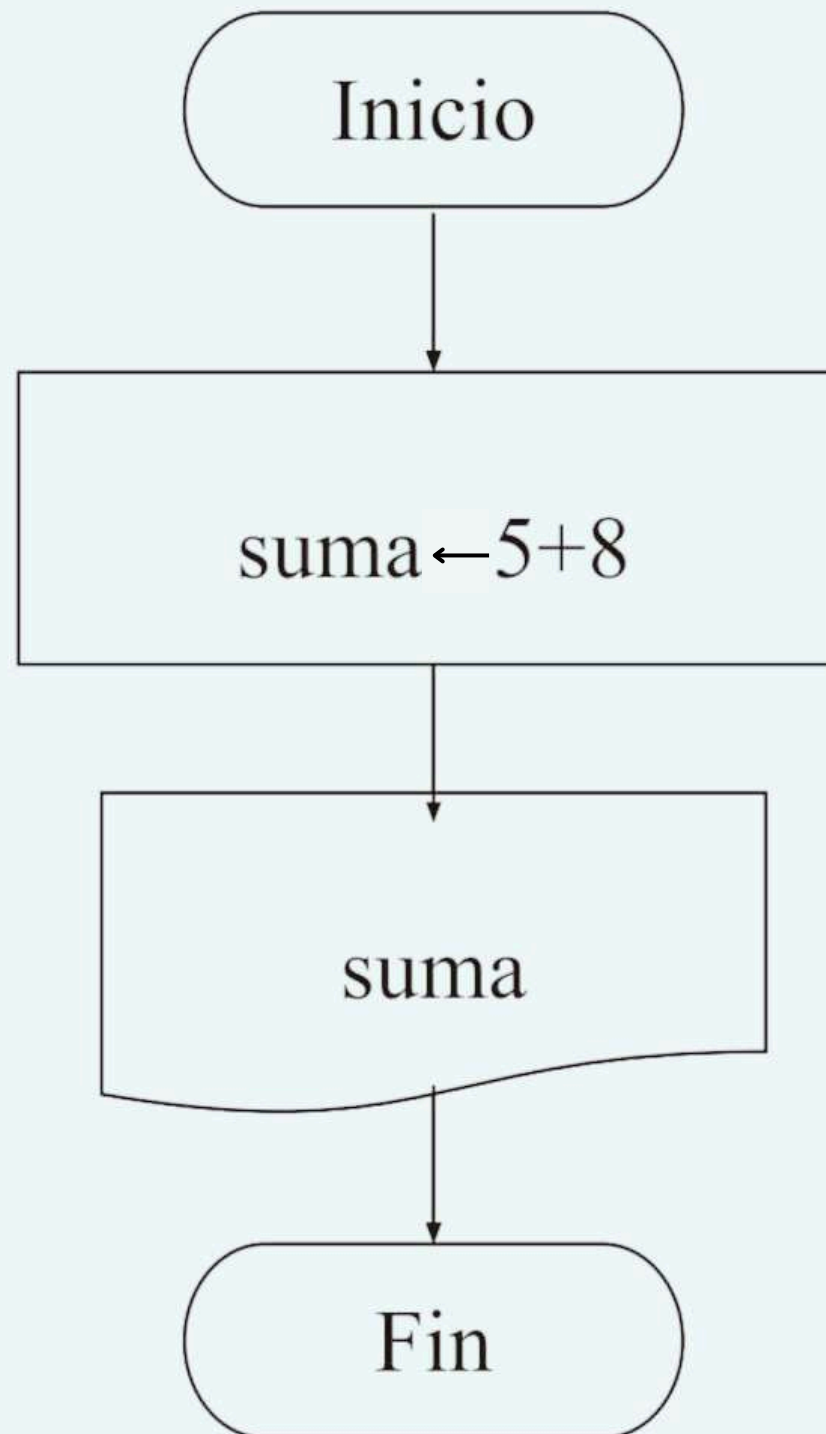
=



Importante
jerarquía de
operadores

EJEMPLO

Obtener la suma de
dos constantes de
tipo numérico



INICIO

REALIAR SUMA

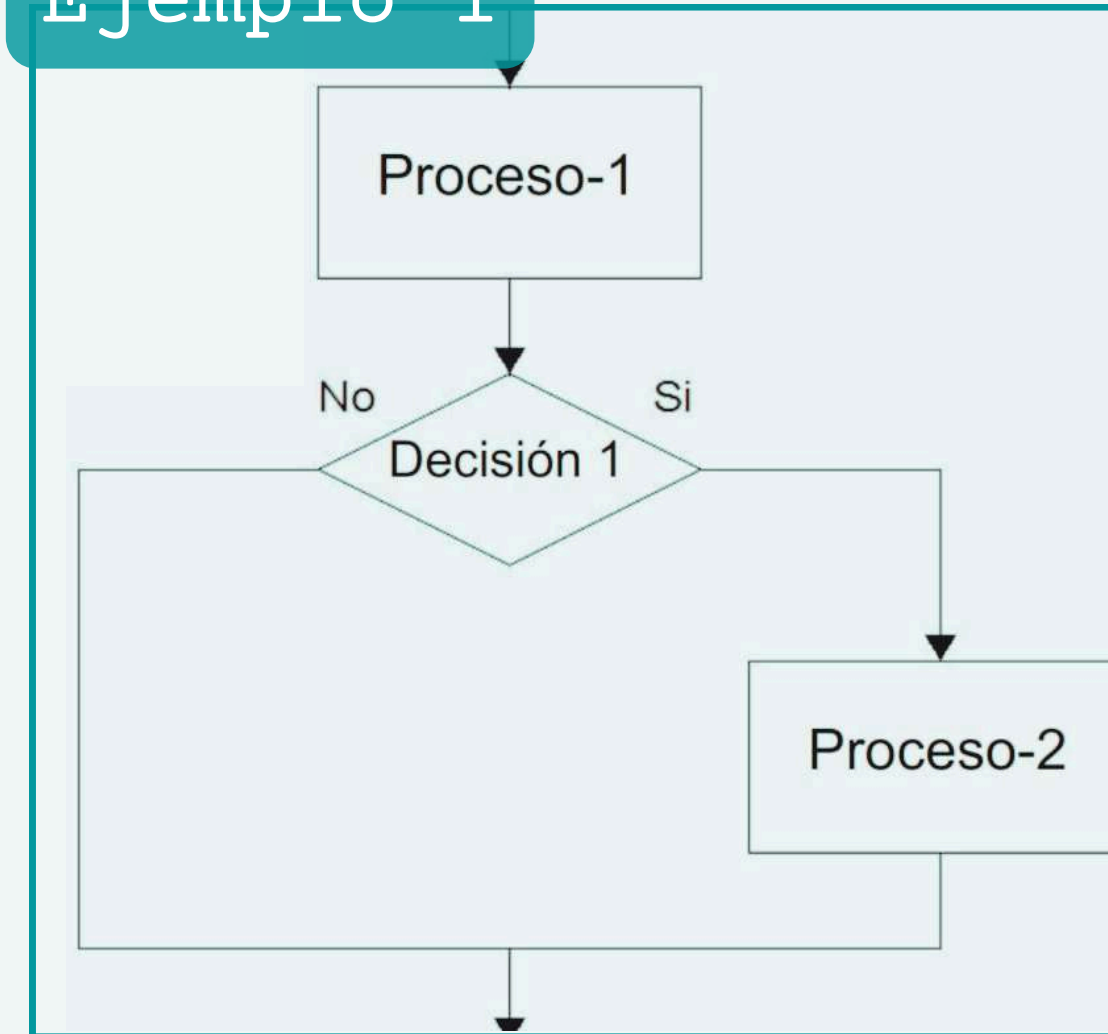
MOSTRAR RESULTADO
SUMA

FIN

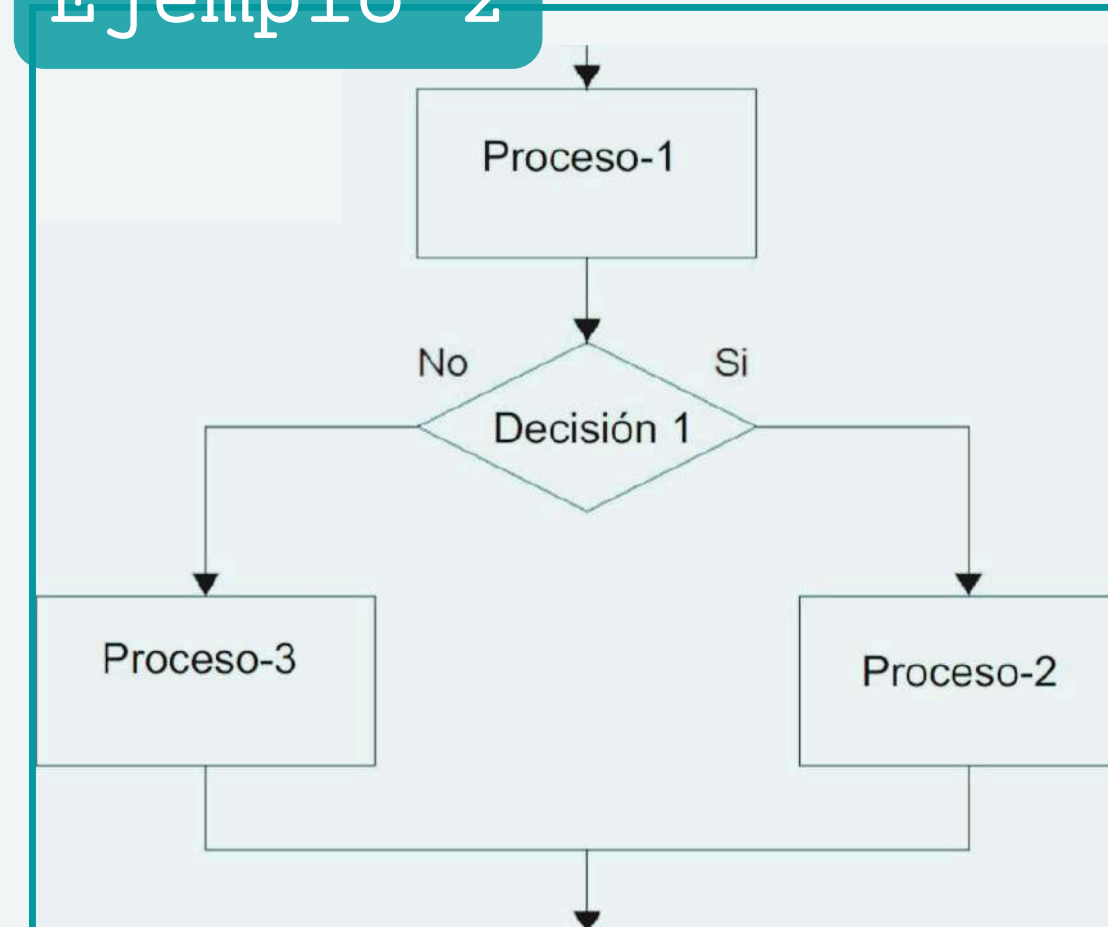
DECISIONES EN UN DIAGRAMA

Representa un punto donde el flujo puede tomar dos caminos dependiendo de la condición

Ejemplo 1



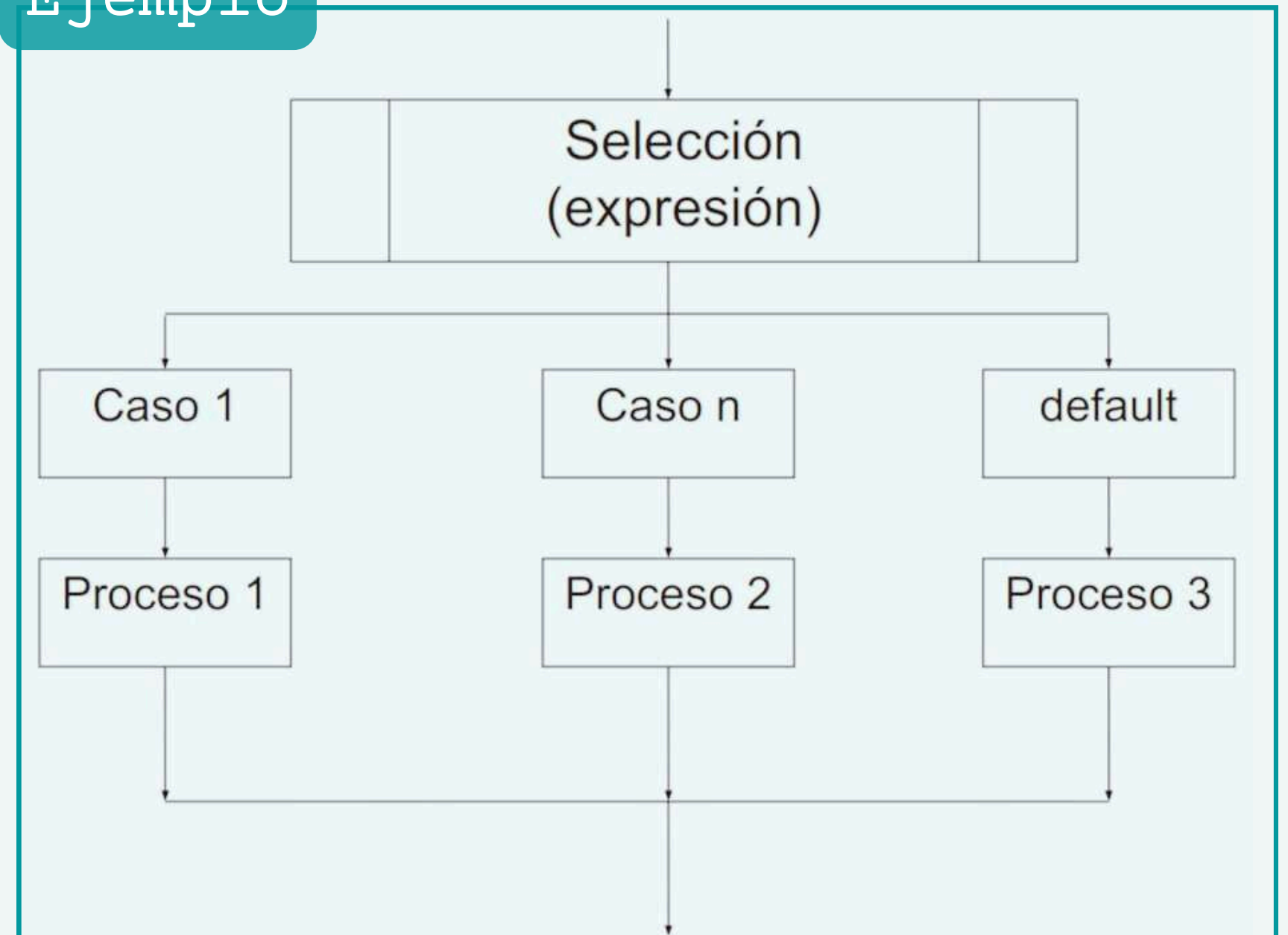
Ejemplo 2



ESTRUCTURAS DE SELECCIÓN MÚLTIPLE

Se usa cuando una variable o condición puede tomar varios valores posibles, y según el valor se elige un camino distinto.

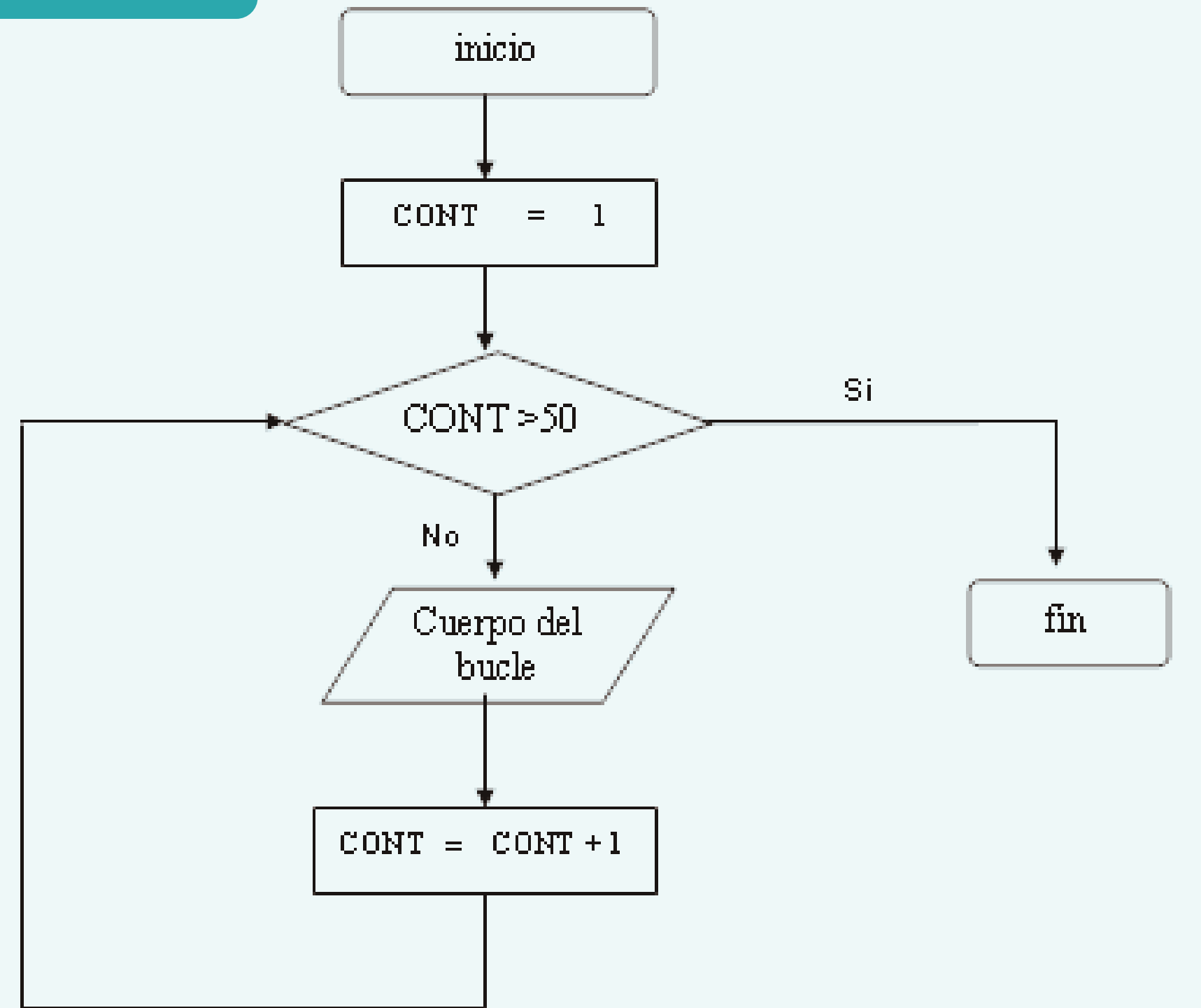
Ejemplo



ESTRUCTURAS DE REPETICIÓN:

Sirven para ejecutar un conjunto de instrucciones varias veces hasta que se cumpla (o deje de cumplirse) una condición.

Ejemplo



PSEUDOCÓDIGO



¿QUE ES?

Representación informal de un algoritmo, escrito en lenguaje natural estructurado que imita la lógica del código pero sin seguir las reglas estrictas de la sintaxis de un lenguaje de programación.

Proceso Suma

Definir A,B,C como Reales;

Escribir "Ingrese el primer numero:";
Leer A;

Escribir "Ingrese el segundo numero:";
Leer B;

C <- A+B;

Escribir "El resultado es: ",C;

FinProceso



Utiliza combinación de lenguaje natural y elementos similares al lenguaje de programación.

EJEMPLO.

Obtener la suma de
dos constantes de
tipo numérico

```
1  Nombre del Algoritmo: Suma
2  Variables
3      suma, real
4  Inicio
5      Asignar suma  $\leftarrow$  5 + 8
6      Mostrar suma
7  Fin
```

VARIABLES

Tipo de datos:

Numéricos

- Enteros `5, -12, 100`
- Reales `3.1416, -2.5`

Alfanuméricos

- Caracteres `a, b, c, i, #`

Datos lógicos

- Booleanos `True / False`

Contador

Lleva la cuenta de determinadas acciones que se pueden solicitar durante la resolución de un problema.

Numero de repeticiones o eventos

Acumulador

Almacena la suma de valores de manera progresiva.

Total de una suma acumulada

OPERADORES LÓGICOS Y RELACIONALES

Relacionales

Menor	<
Mayor	>
Igual	=
Menor e igual	<=
Mayor e igual	>=
Diferente	!=

Lógicos

AND

OR



Comparación de tres variables
A = B = C (incorrecto)

A = B y B = C correcto

A = B o B = C o A = C correcto

DECISIONES PSEUDOCÓDIGO

Tomar distintos caminos
según una condición dada
(verdadero / falso)

Ejemplo 1

ASIGNAR PROCESO 1
SI DECISIÓN 1 (ES VERDADERA) ENTONCES
ASIGNAR PROCESO 2
FIN DE DECISION

Ejemplo 2

ASIGNAR PROCESO 1
SI DECISIÓN 1 (ES VERDADERA) ENTONCES
ASIGNAR PROCESO 2
SI NO
ASIGNAR PROCESO 3
FIN DE DECISION

ELEMENTOS SELECCIÓN MÚLTIPLE PSEUDOCÓDIGO

Variable puede tomar
varios valores y según
el caso se ejecuta una
acción distinta

Ejemplo 1

SI SELECTOR IGUAL

VALOR 1: HACER ACCIÓN 1

VALOR 2: HACER ACCIÓN 2

VALOR 3: HACER ACCIÓN 3

VALOR 4: HACER ACCIÓN 4

DE OTRA FORMA: HACER ACCIÓN N
FIN DEL CONDICIONAL (SELECTOR)

ESTRUCTURAS DE REPETICIÓN

Ejecutar un bloque de
código varias veces
dependiendo de una
condición o contador

Ejemplo 1

Nombre de algoritmo: ciclo1

Variables: X, entero

Inicio

asignar $X \leftarrow 1$

mientras $X \leq 10$ repetir

mostrar X

asignar $X \leftarrow X + 1$

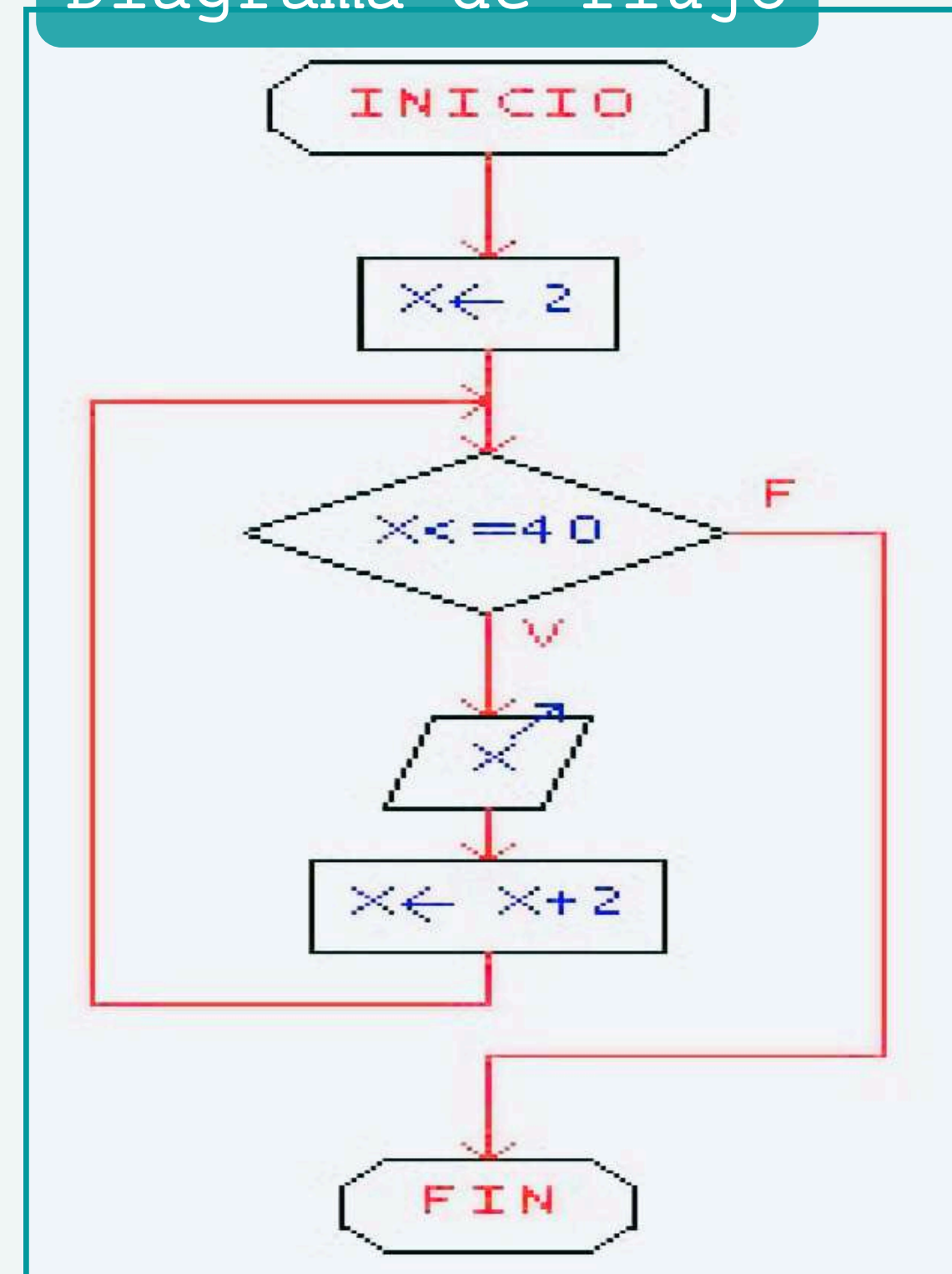
fin ciclo

Fin

EJEMPLO.

Generar e imprimir los primeros 20 números pares. Usar una variable y un ciclo mientras

Diagrama de flujo



Pseudocódigo

Nombre del algoritmo: Ciclo2

variables: X, entera

Inicio

Asignar $X \leftarrow 2$

mientras $X \leq 40$ repetir

mostrar X

asignar $X \leftarrow X + 2$

fin de ciclo

Fin

REFERENCIAS

- Metodología de la programación : algoritmos, diagramas de flujo y programas / Osvaldo Cairó Battistutti (3a. edición). (2005). Alfaomega.
- Patiño Cortés, I., Carrasco Vallinot, R., & Raya Cabrera, J. L. (2006). Fundamentos de programación / Manuel Santos González, Ismael Patiño Cortés, Raúl Carrasco Vallinot ; coordinador, José Luis Raya Cabrera. Alfaomega.