

# Pensamiento computacional y lógica pura

## ATENCIÓN · Regla de la semana: ● prohibido usar IA

Esta semana **no hay computador encendido para programar**. Hay tablero, papel y tu cabeza. Punto. Queremos que entiendas cómo «piensa» la máquina *antes* de que una IA piense por ti.

- **Rojo.** IA prohibida. Esta es la semana del semáforo en rojo.
- **Amarillo.** (más adelante) IA solo como tutora de tus errores.
- **Verde.** (más adelante) IA como copiloto, contigo al mando.

## LO ESENCIAL · El objetivo

Al terminar serás capaz de **ejecutar un algoritmo a mano**, paso a paso, como si tú fueras el procesador. Esa técnica se llama **prueba de escritorio** (*desk-check*) y es la habilidad #1 de todo programador serio.

## DEFINICIÓN · El algoritmo

Una secuencia de pasos **ordenados, claros y finitos** para resolver un problema.

- 1 **Ordenado** — el orden cambia el resultado (primero medias, luego zapatos).
- 2 **Claro** — sin ambigüedad. La máquina no adivina.
- 3 **Finito** — termina. No da vueltas para siempre.

*¿por qué? si no sabes hacia dónde va el código, no puedes notar cuándo la IA te entrega basura disfrazada de oro.*

## DEFINICIÓN · Estructuras de datos: lista y diccionario

La forma en que guardamos la información. Dos que usarás toda la vida:

- L Lista** — una fila ordenada de cosas. `frutas = ["mango", "lulo", "guayaba"]`. Se cuenta desde 0: el primero es la posición 0.
- D Diccionario** — pares **clave** → **valor**, como un directorio. `edades = {"Ana": 14}`. No cuentas posiciones: preguntas por el nombre. `edades["Ana"] → 14`.

**DEFINICIÓN** · Control de flujo: decidir y repetir

Cómo la máquina elige qué hacer y cuándo.

**?** **Condiciona** (`if / else`) — DECIDIR. SI la nota  $\geq 3.0$ , escribe «Aprobó»; SI NO, escribe «Reprobó».

**🕒** **Bucle** (`for / while`) — REPETIR. PARA cada estudiante en la lista, escribe su nombre. Una instrucción, no treinta.

**NOTA** · **🌐 Tech English**

*algorithm* = algoritmo · *step* = paso · *input* = lo que entra · *output* = lo que sale · *list* = lista · *dictionary* (dict) = diccionario · *key* = clave · *value* = valor · *index* = índice · *if* = si · *else* = si no · *loop* = bucle · *for each* = para cada · *to trace* = seguir el código a mano · *counter* = contador · *greater than* = mayor que.

**Ejemplo 1. Prueba de escritorio resuelta**

Sigue este algoritmo a mano y anota cómo cambian los valores:

**total = 0** *empieza el acumulador en cero*

**PARA** cada precio en [2, 5, 3]: *recorre la lista, uno por uno*

**total = total + precio** *suma el precio actual al total*

**escribe total** *muestra el resultado al final*

Paso	precio	total
inicio	—	0
1	2	$0 + 2 = 2$
2	5	$2 + 5 = 7$
3	3	$7 + 3 = 10$

**Resultado:** escribe **10**. Seguiste a la máquina sin computador. Esa es la habilidad: el que sabe hacer esto, sabe programar. El resto es vocabulario.

**PRÁCTICA** · A mano, sin computador y sin IA

● Básico

1. Escribe, en pasos numerados y en español claro, el algoritmo para **preparar un vaso de jugo de mango**. Mínimo 6 pasos; cada paso, *una sola acción*. Imagina que lo lee un robot que toma TODO literal (si no dices «pela el mango», se lo come con cáscara). ¿Funcionaría al pie de la letra? ¿Falta algún paso «obvio» que el robot no adivinaría? [*orden y precisión*]

● Intermedio

2. Una entrada al cine cuesta \$10.000. Si la persona es estudiante, paga \$7.000. Y si además es miércoles, todos pagan \$5.000 sin importar nada más. (a) Escribe el pseudocódigo con las decisiones necesarias. (b) Calcula el precio para: persona normal en martes; estudiante en martes; persona normal en miércoles. Pista: el miércoles «manda» sobre todo lo demás — ¿tu lógica lo refleja? [*condicionales*]

● Reto

3. Diseña en pseudocódigo un algoritmo que, dada una lista de números, **encuentre el más grande**, mirando *un número a la vez* (como pasar cartas y recordar «la más grande hasta ahora»). ¿Funcionaría con cualquier lista, incluso una de 1.000 números? ¿Qué pasa con el primer número? [*diseño lógico*]

## ✂ HOJA DE TRABAJO · Prueba de escritorio — llénala a mano

Nombre: \_\_\_\_\_

Fecha: \_\_\_\_\_

Sigue este algoritmo **paso a paso** (no lo «corras» en ningún lado) y llena la tabla. Marca en cada fila si el número es mayor que 5 y cómo queda el contador.

```

contador = 0
PARA cada numero en [4, 7, 2, 9]:
  SI numero es mayor que 5:
    contador = contador + 1
  escribe contador

```

Paso	numero	¿mayor que 5?	contador
inicio	—	—	0
1	4		
2	7		
3	2		
4	9		

**Pregunta final:** ¿qué número escribe el programa? \_\_\_\_\_

*Criterio:* esta es LA habilidad de la semana. La tabla bien llena = dominaste el *desk-check*.

#### LO ESENCIAL · La idea que te debes llevar

Programar no es escribir en un teclado. Es **pensar con precisión brutal**. El teclado vino después; el pensamiento va primero. Equivocarte *pensando* es justo lo que entrenamos esta semana. Equivocarte *copiando* no enseña nada — y esta semana no hay de dónde copiar.