

# CURSO DE FUNDAMENTOS DE PROGRAMACIÓN DESDE CERO



# ¿CÓMO ANALIZAR UN PROBLEMA ANTES DE ESCRIBIR EL ALGORITMO?



# CÓMO ANALIZAR UN PROBLEMA?

Analizar un problema es el primer paso en la creación de un programa.



# CÓMO ANALIZAR UN PROBLEMA?

Consiste en comprender qué se necesita resolver, qué información se tiene y qué resultado se espera.



# CÓMO ANALIZAR UN PROBLEMA?

Un buen análisis evita errores, simplifica el diseño del algoritmo y mejora la eficiencia del programa.



# 1. COMPRENDER EL PROBLEMA

Antes de pensar en pasos o soluciones, es fundamental entender con claridad qué se pide.



# 1. COMPRENDER EL PROBLEMA

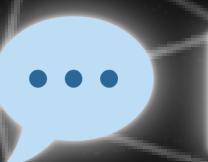
PREGÚNTATE:

¿Qué problema necesito resolver?

¿Cuál es el objetivo del programa?

¿Qué resultado quiero obtener?

# 1. COMPRENDER EL PROBLEMA



EJEMPLO:

Si el problema es “calcular el promedio de tres notas”, el objetivo es obtener el promedio de tres números dados.

## 2. IDENTIFICAR LAS ENTRADAS, PROCESOS Y SALIDAS

Todo problema puede analizarse dividiéndolo  
en tres partes:





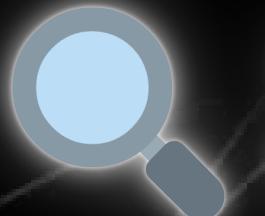
## 2. IDENTIFICAR LAS ENTRADAS, PROCESOS Y SALIDAS

Elemento	Descripción	Ejemplo (promedio de 3 notas)
Entradas	Datos que se reciben	nota1, nota2, nota3
Proceso	Operaciones necesarias	sumar y dividir entre 3
Salida	Resultado final	promedio

## 2. IDENTIFICAR LAS ENTRADAS, PROCESOS Y SALIDAS

👉 Este análisis ayuda a visualizar qué información necesitas y cómo se transformará.

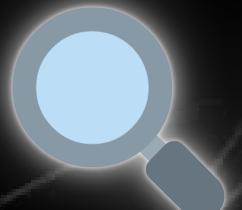




### 3. ANALIZAR LAS CONDICIONES Y RESTRICCIONES

Muchos problemas tienen reglas o limitaciones que deben cumplirse.





### 3. ANALIZAR LAS CONDICIONES Y RESTRICCIONES

Por ejemplo:

¿Qué hacer si una nota es negativa?

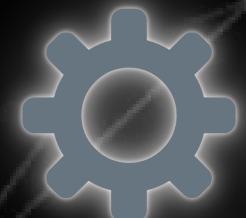
¿Hay un valor máximo permitido?

¿Qué sucede si el usuario no introduce datos?

# 3. ANALIZAR LAS CONDICIONES Y RESTRICCIONES

Este paso evita errores lógicos más adelante y ayuda a crear algoritmos robustos.

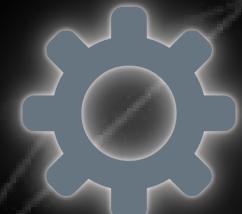




## 4. DIVIDIR EL PROBLEMA EN PARTES (MODULARIZACIÓN)

Un problema grande puede ser difícil de resolver de una sola vez.

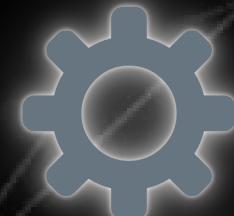




## 4. DIVIDIR EL PROBLEMA EN PARTES (MODULARIZACIÓN)

Por eso, es útil dividirlo en subproblemas más pequeños.





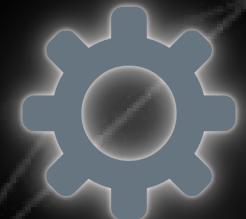
## 4. DIVIDIR EL PROBLEMA EN PARTES (MODULARIZACIÓN)

Ejemplo: Para calcular el promedio de un grupo de estudiantes:

Subproblema 1: Leer las notas de cada estudiante.

Subproblema 2: Calcular el promedio individual.

Subproblema 3: Mostrar los resultados.



## 4. DIVIDIR EL PROBLEMA EN PARTES (MODULARIZACIÓN)

Esta técnica se llama modularización y facilita la organización del algoritmo.





# 5. VERIFICAR LA LÓGICA CON EJEMPLOS

Antes de escribir el algoritmo, se recomienda probar mentalmente la solución con valores reales.





# 5. VERIFICAR LA LÓGICA CON EJEMPLOS



EJEMPLO:

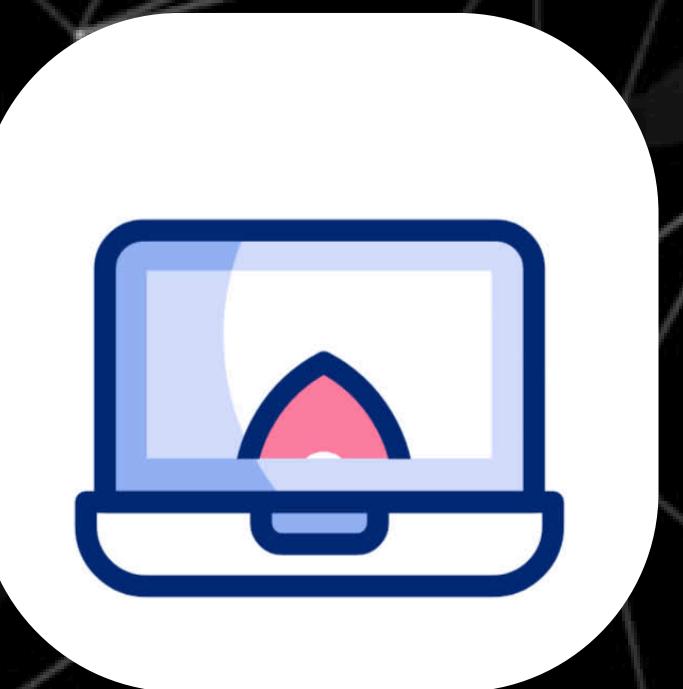
Si las notas son 10, 8 y 6 →

$(10 + 8 + 6) / 3 = 8 \rightarrow$  El promedio es correcto.



# 5. VERIFICAR LA LÓGICA CON EJEMPLOS

Este paso permite comprobar si el razonamiento es lógico y si todos los pasos están contemplados.





## 6. REDACTAR EL ALGORITMO O PSEUDOCÓDIGO

Una vez analizado el problema, se puede pasar a escribir el algoritmo en forma de pseudocódigo o diagrama de flujo, sabiendo con claridad qué se debe hacer.





# CONCLUSIÓN

Analizar un problema antes de escribir el algoritmo es pensar como un programador





# CONCLUSIÓN

Es importante: comprender, organizar y planificar antes de actuar.





# CONCLUSIÓN

Este proceso asegura que el algoritmo sea correcto, eficiente y fácil de implementar, evitando errores comunes y ahorrando tiempo en la programación.



**NOS VEMOS EN UN PRÓXIMO  
VIDEO DE ESTE CURSO,  
SALUDOS**

