

# Introducción a la lógica de programación

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Resolución de Algoritmos

**DEV.F**  
DESARROLLAMOS(PERSONAS);

dev

# Lógica y programación

Vamos a abordar dos grandes conceptos, **lógica** y **programación**.

La **lógica** es la ciencia de relaciones que estudia el pensamiento, pero esta se enfoca en el estudio de la relaciones mismas (como en una ciencia formal) y la **programación** es el proceso de tomar un algoritmo y codificarlo con alguna notación existente.

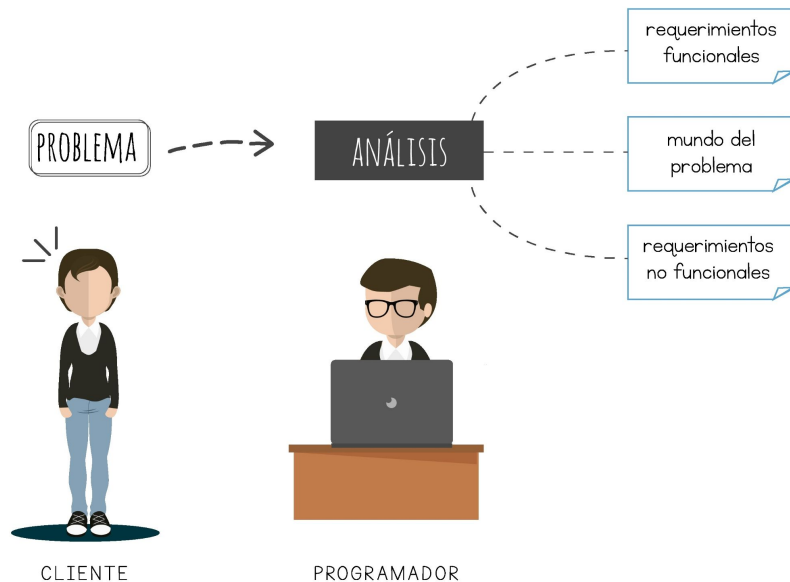


# Lógica y programación

“La programación es el arte de resolver problemas a través del uso de la tecnología.”

La lógica = el pensamiento creativo.

La programación = el medio para llegar a la solución del problema.



# ¿Qué tan creativo eres?

el programador cuenta con un conjunto de herramientas y lenguajes para construir la solución



PROGRAMADOR

HERRAMIENTAS Y LENGUAJES

análisis del  
problema

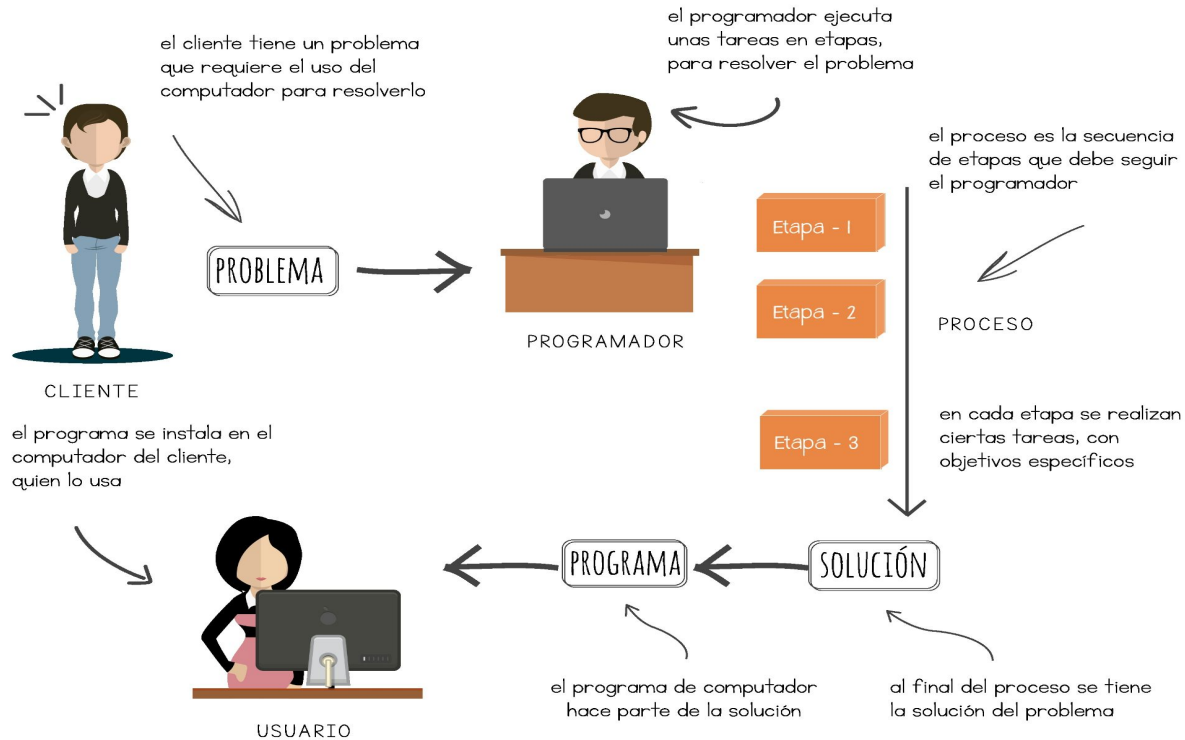
diseño de la  
solución

construcción  
de la solución

PROBLEMA

proceso

SOLUCIÓN



# Análisis de problema

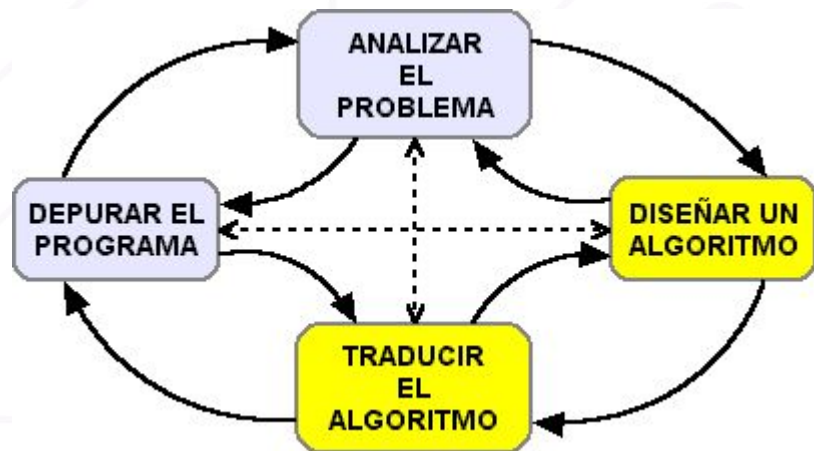
## Entrada

- Fuentes de información
- Variables
- Constantes

## Salida

- Fuentes de información
- Variables

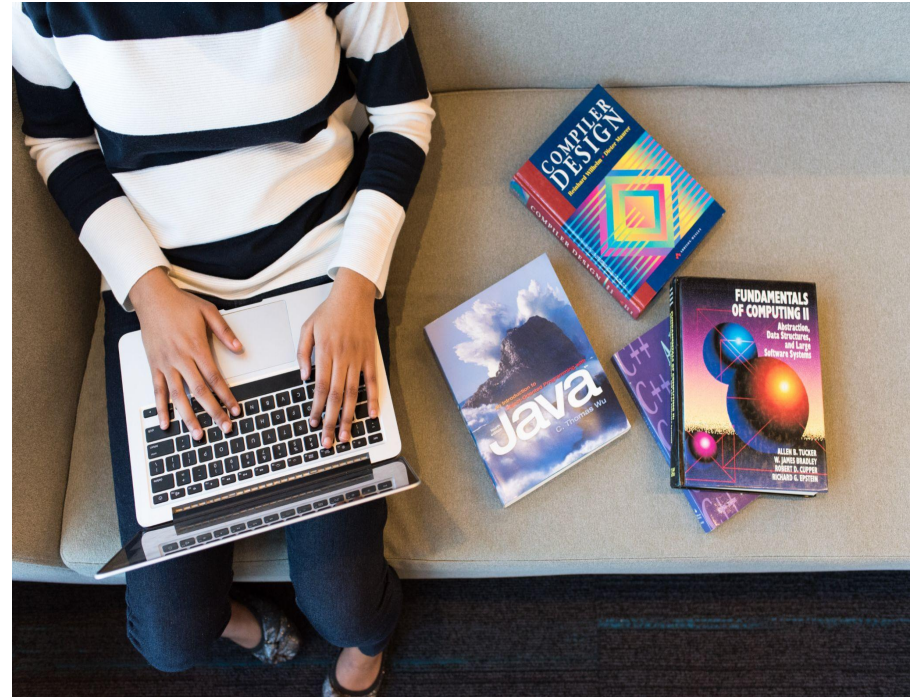
# Enfoque del problema





# Algoritmo

Es la secuencia de pasos, de reglas o instrucciones que nos permite llegar a la solución de un problema y es la base de la programación.



# Partes de un algoritmo

- ENTRADAS
- PROCESOS
- SALIDAS

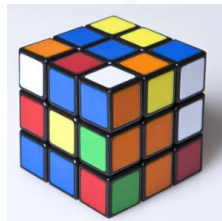


# Partes de un algoritmo

**Entrada:** Son los datos que se le dan al algoritmo

$a = 1;$

$b = 2;$

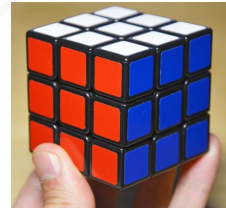


**Proceso:** Son las operaciones que se hacen con los datos

$\text{Suma} = a + b;$

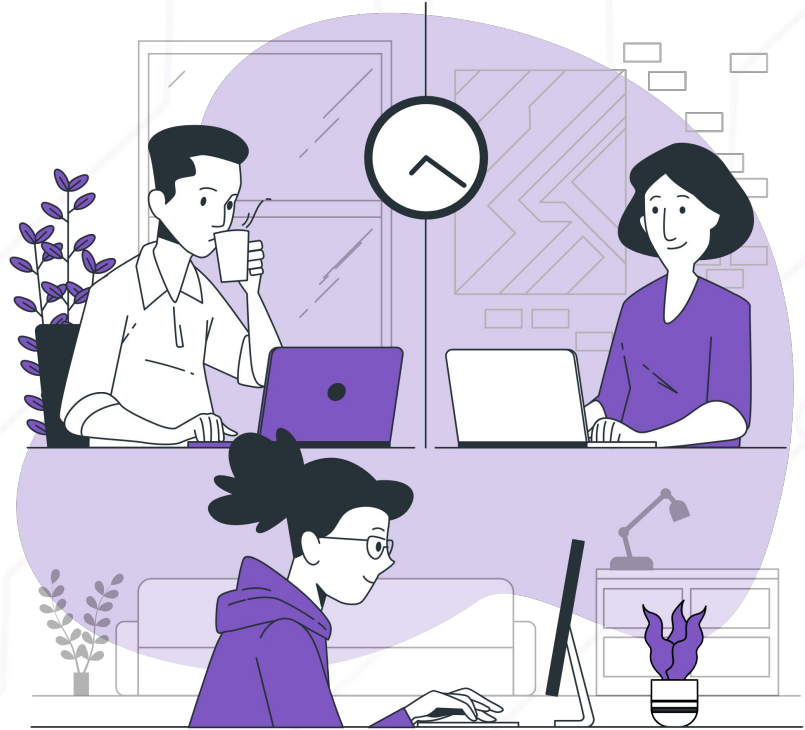
**Salida:** Es el resultado final que se obtiene de las operaciones,

en este caso será 3  
`document.write(Suma)`  
`console.log(Suma)`



# Características

- PRECISO
- DEFINIDO
- FINITO
- LEGIBLE

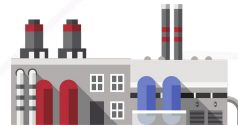


# Características

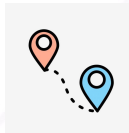
**Preciso:** Tiene que resolver el problema sin errores



**Definido:** Si ejecutas el algoritmo varias veces los datos de salida serán iguales en cada repetición



**Finito:** Debe tener un inicio y un final.

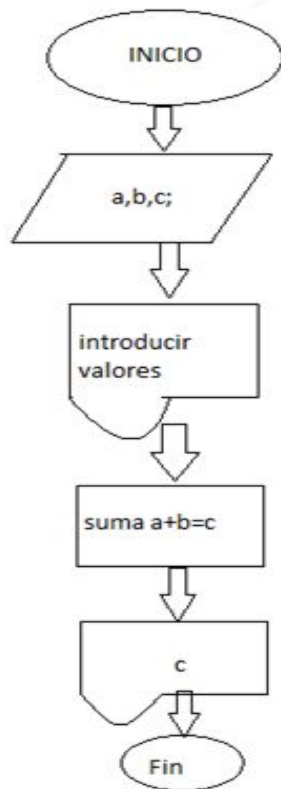


**Legible:** Cualquier persona que vea el algoritmo debe ser capaz de comprenderlo.



# Formas de representar un algoritmo

DDF



**algoritmo** Sumar

**variables**

entero a, b, c

**inicio**

escribir( "Introduzca el primer número (entero): " )

leer( a )

escribir( "Introduzca el segundo número (entero): " )

leer( b )

$c \leftarrow a + b$

escribir( "La suma es: ", c )

**fin**

Pseudocódigo

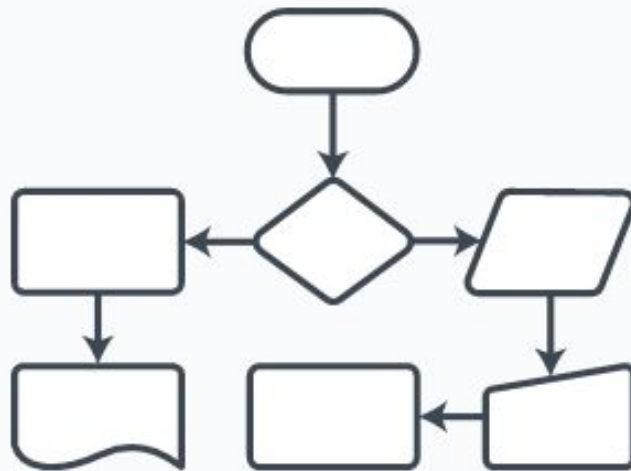
Lenguajes  
De programación



# Diagramas de Flujos (DDF)

Un diagrama de flujo es un **diagrama** que describe un proceso, sistema o algoritmo informático.

Se usan ampliamente en numerosos campos para **documentar, estudiar, planificar, mejorar y comunicar procesos** que suelen ser complejos en diagramas claros y fáciles de comprender.



# Símbolos de diagramas de flujo

Terminal/Terminador

Terminator

Proceso

Process

Documento

Document

Decisión

Decision

Datos o entrada/salida

Data

Datos almacenados

Database

Flecha de flujo



Comentario o anotación



Proceso predefinido

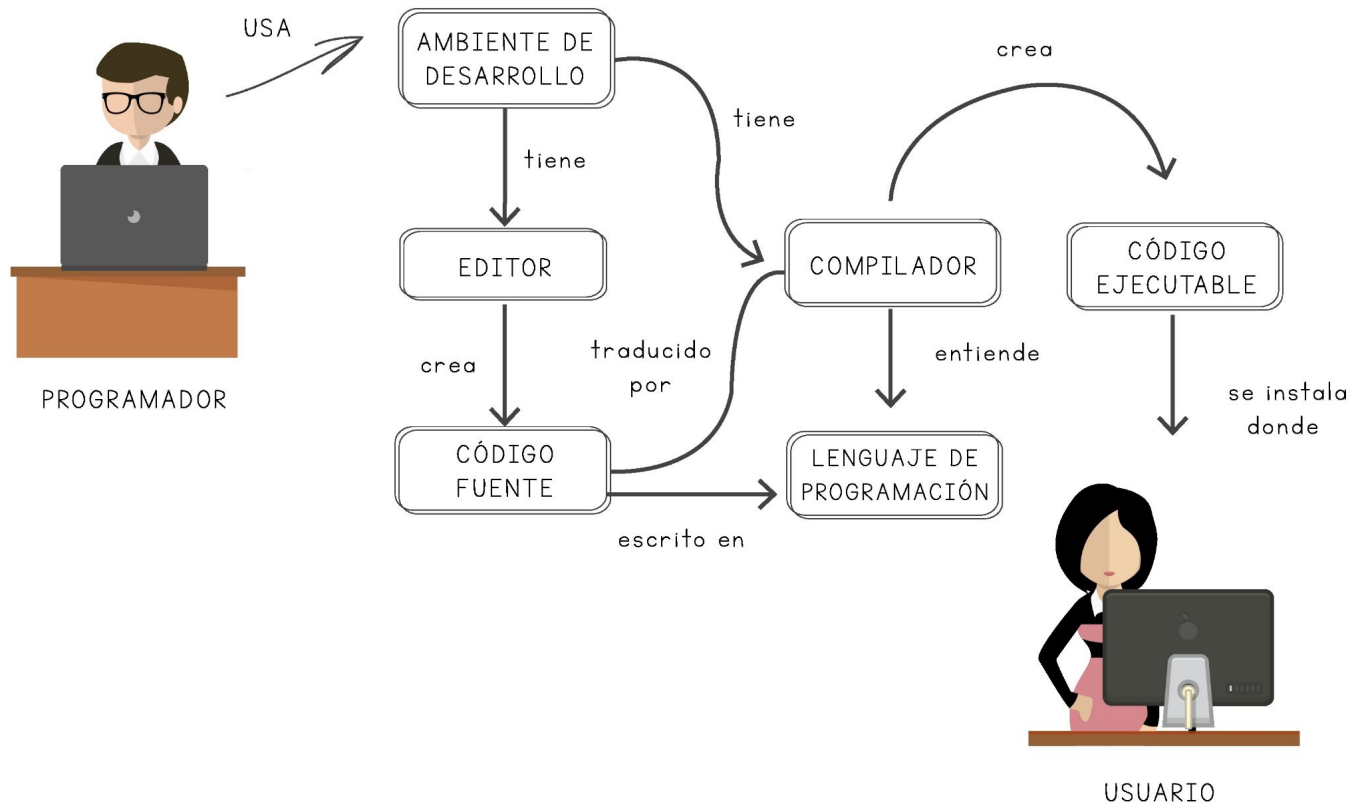
Predefined  
Process

Referencia/conector dentro de la página

A



# DDF EN EL TRABAJO DEL DÍA A DÍA



# Diagramas de Flujos en la vida real



# Pseudocódigo

## Definición Técnica:

En ciencias de la computación, y análisis numérico, el pseudocódigo **es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.**

## Dicho en otras palabras:

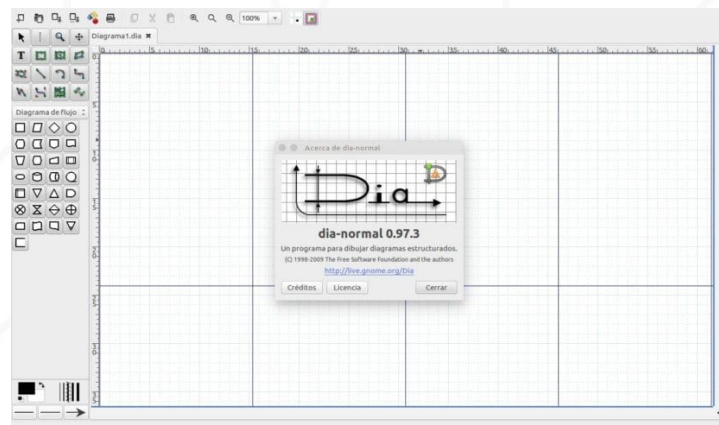
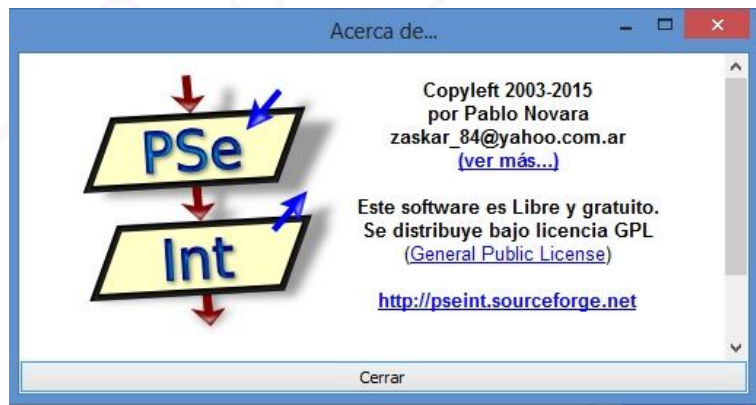
El pseudocódigo es una forma de **escribir los pasos que va a realizar un programa de la forma más cercana al lenguaje de programación que vamos a utilizar posteriormente.**

Es como un falso lenguaje, pero es muy útil en nuestro idioma (Idioma Humano)

# Ejemplo

```
1  Proceso NUMERO_MAYOR
2      Escribir 'Ingresa A: ';
3      Leer A;
4      Escribir 'Ingresa B: ';
5      Leer B;
6      Si A > B Entonces
7          Escribir 'El mayor es A';
8      Sino
9          Escribir 'El mayor es B';
10     FinSi
11 FinProceso
12
```

# Software



**bizagi**  
Studio



# Plataformas para estimular la lógica

HackerRank

Login

Sign Up

Matching developers  
with great companies.

## For Companies

We are the market-leading technical interview platform to identify and hire developers wherever they are.

Start Hiring

## For Developers

Join over 11 million developers, practice coding skills, prepare for Interviews and get hired.

Sign Up & Code



<https://www.hackerrank.com/>

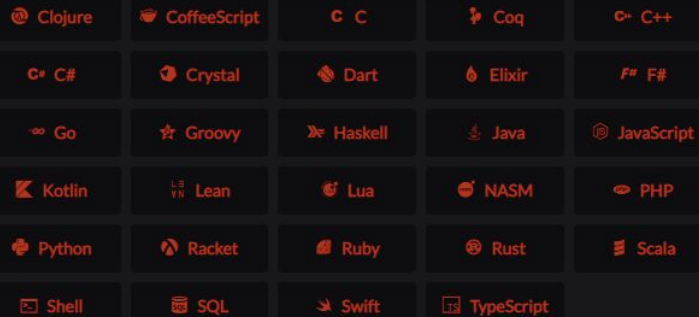


## Achieve mastery through challenge

Improve your skills by training with others on real code challenges

SIGN UP

To join you must first prove your skills.  
Choose your language to begin...



Additional Languages



<https://www.codewars.com/>

\*These languages are currently in beta. Once you enlist you will have an opportunity to train with them.

## ¿Cómo puedo mejorar mi lógica de programador?

1. **Intenta entender el código de otros programadores.** Cuando trabajes en equipo o aprendas por tu cuenta. Al ver el código de otros programadores, descubres formas distintas de solucionar un mismo problema, formas que quizás a ti nunca se te hubiera ocurrido.
2. **Resuelve algoritmos.** Un algoritmo es la forma que tenemos para solucionar un problema sin la necesidad de programarlo. Soluciona tus ejercicios de programación en un papel, antes de intentar llevarlo a código.
3. **Aprende pseudocódigo.** El pseudocódigo es una forma de estructurar tu algoritmo en un lenguaje intermedio. No llega a ser un lenguaje de programación, pero tampoco es el lenguaje común con el que hablamos todos los días.
4. **Programa mucho.** No intentes desarrollar de frente programas grandes; sino, empieza desarrollando cientos de ejercicios pequeños.