



## Segunda entrega:Glosario

Emanuel Barboza García

Universidad CENFOTEC

Principios de Programación 1

Kevin Jiménez Garro

Fecha: Noviembre, 2025

## **Glosario**

### **Programa**

Es un conjunto de instrucciones escritas en un lenguaje de programación para que una computadora pueda ejecutarlas.

Ejemplo:

- print("Hola Mundo")

### **Algoritmo**

Son una serie de pasos lógicos y ordenados destinados a realizar una tarea en específico o resolver un problema

Ejemplo:

- Pasos para lavarse los dientes:
- 1-abrir la pasta
- 2-echar la pasta en el cepillo.
- 3-cebillarse los dientes
- 4-Enjuagarse la boca con agua
- 5-Secarse

### **Compilador**

El compilador toma el código, revisa si hay errores y después lo pasa a lenguaje de máquina en un archivo ejecutable para que la computadora pueda interpretarlo y ejecutarlo.

Ejemplo de lenguajes que utilizan compiladores:

- C
- C++
- GO

### **Intérprete**

El intérprete al vez de revisar el código y después traducirlo a lenguaje de máquina como lo hace el compilador, el intérprete va ejecutando línea por línea y lo va traduciendo a lenguaje de máquina sin generar un archivo ejecutable.

Ejemplo de lenguajes que utilizan compiladores:

- Python
- Javascript

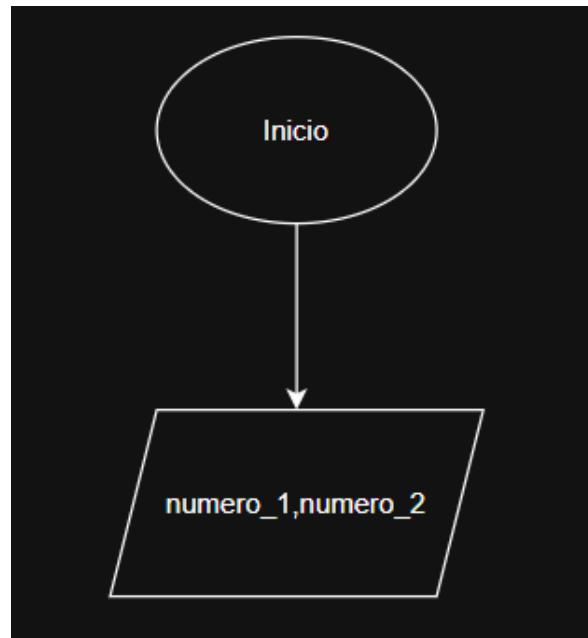
### **Diagrama de Flujo**

Es la forma gráfica de representar una serie de instrucciones de un algoritmo con formas específicas que significan por ejemplo las entradas, procesos, salidas.

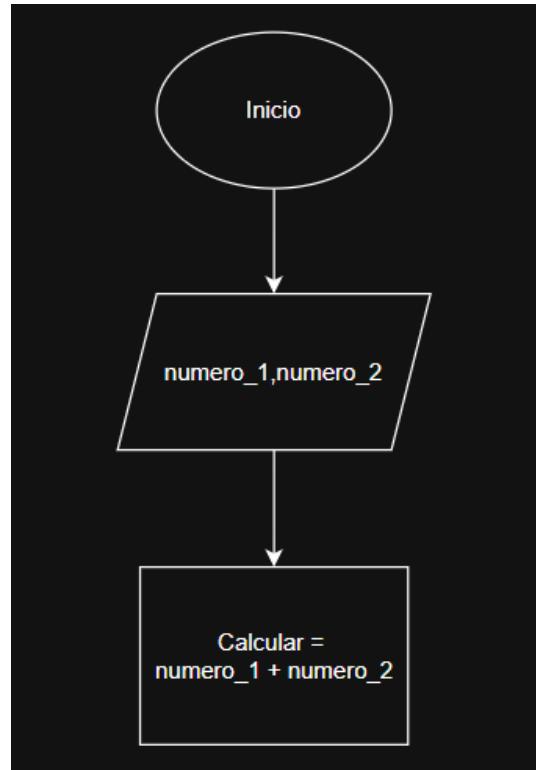
## Pseudocódigo

Son la serie de instrucciones en lenguaje normal para poder entender y cómo se van estructurar con las ideas para ejecutar el código.

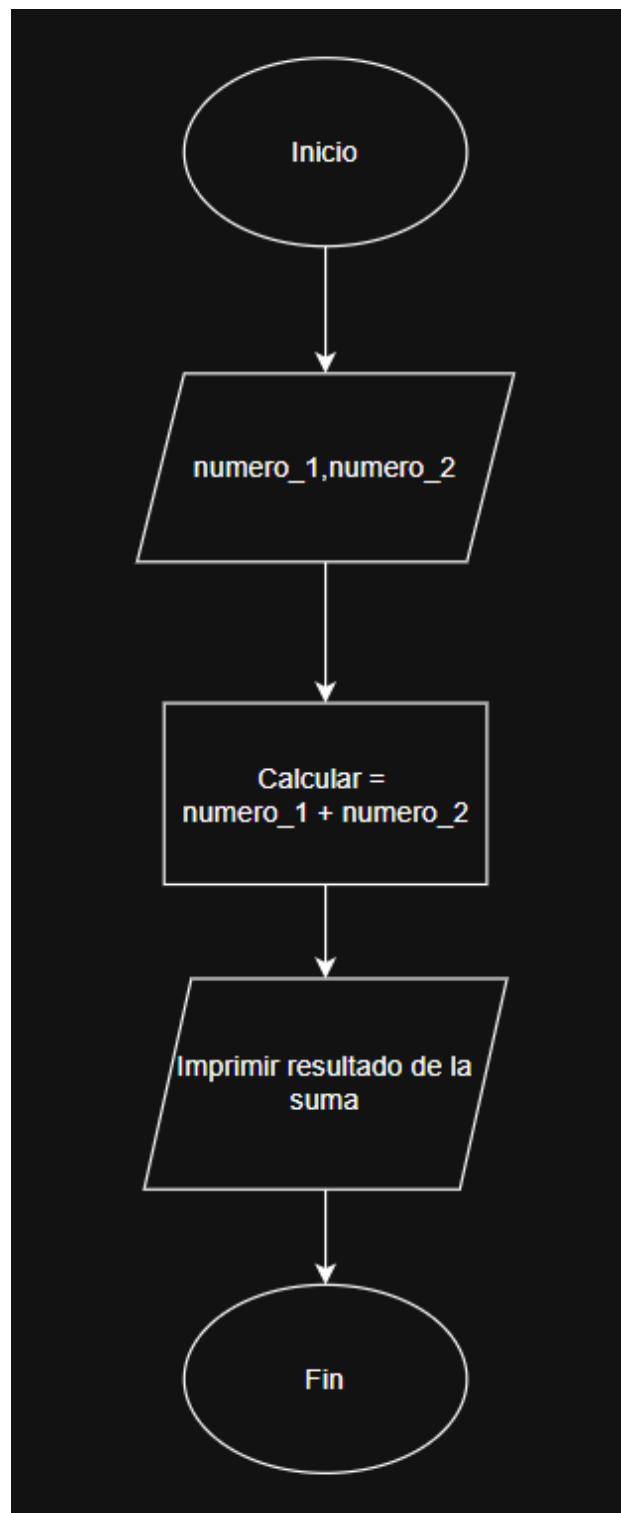
### Entrada de datos(imagen)



### Asignación o proceso(imagen)



## Salida de datos(imagen)



## **Variable**

La variable almacena un valor en un espacio de la memoria. El valor que se almacena puede variar a lo largo del programa dependiendo de lo que se ejecute.

### **Asignación de variable**

Es el nombre que le colocamos al espacio de almacenamiento(variable) para identificarlo y diferenciarlo del resto.

Ejemplos:

- nombre\_1
- numero\_2
- edad

### **Tipo de dato de una variable**

Como el nombre lo indica es el tipo de valor que se le puede asignar a una variable.

Ejemplo:

- Int - Valores enteros: 1,2,3,4
- Float - Valores decimales: 1.1, 70.5
- Str - Caracteres: hola
- Booleanas: Falso y Verdadero.
- Entre otras...

### **Valor de una variable**

Es el valor que se almacena en una variable, puede contener un dato específico o datos modificables que pueden cambiar en cualquier momento durante el código.

### **Tipos de datos primitivos**

Son los tipos de datos que se le puede asignar a una variable, como caracteres, números enteros, decimales, listas, datos booleanos.

Ejemplo:

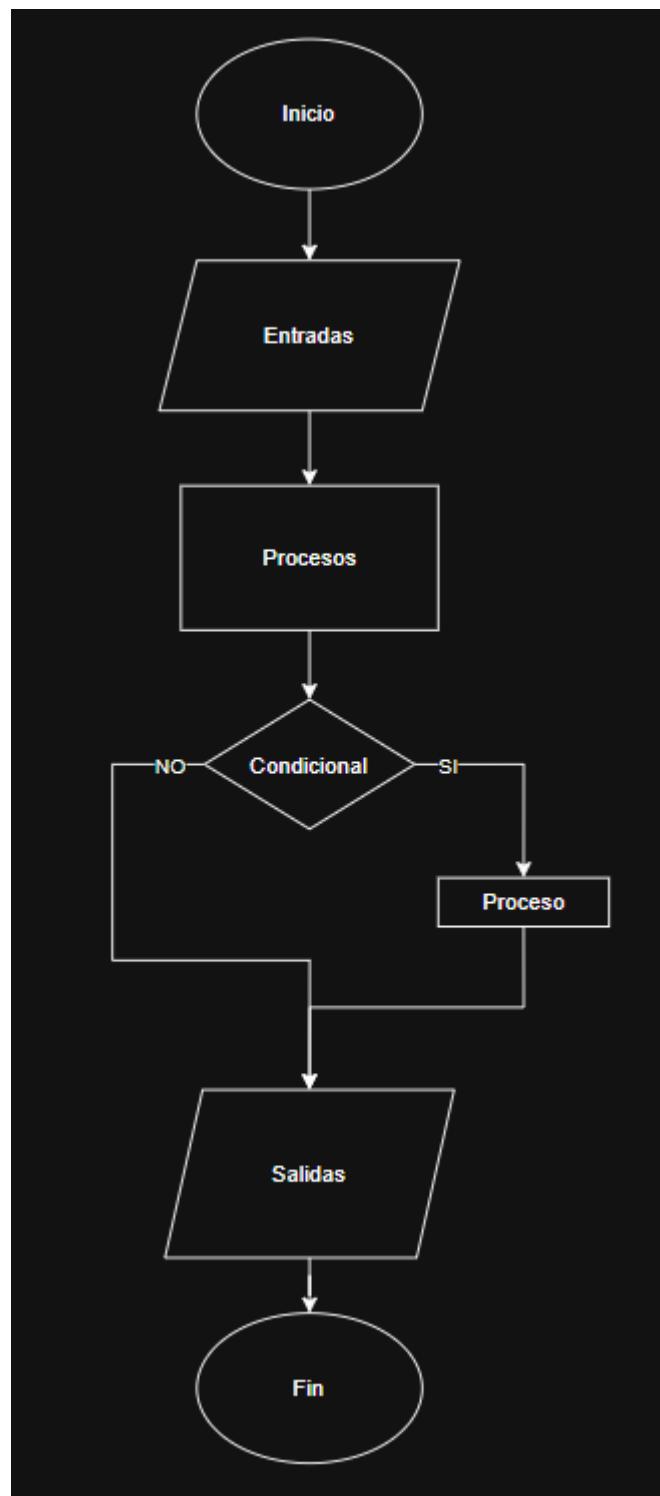
- nota = 89.5
- edad = 26
- nombre: juan

### **Pasos y explicación para la creación de un programa estructurado**

1. Definición del problema: Surge ante alguna necesidad, mejora o descubrimiento.
2. Análisis del Problema: Analizar cada detalle del problema para poder entenderlo y descubrir una solución.

3. Diseño de la solución: Ante el problema analizado, dar las instrucciones paso a paso para resolver el problema(pseudocódigo,diagrama).
4. Implementación: Desarrollar código para la solución del problema dado.
5. Control y calidad: Realizar la pruebas correspondientes para validar que el programa desarrollado funciona bien y soluciona el problema.

### Estándar de inicio y fin en los diagramas de flujo



## Categorías de variables

- Variables locales: Se ejecutan solo dentro de una función.
- Variables globales: se pueden usar en todo el programa.

## Tabla de variables con ejemplos

Nombre	Tipo	Valor	Categoría
edad	int	20	Global
nombre	str	"Ana"	Local
promedio	float	85.5	Global
activo	bool	True	Local

## Operadores aritméticos

- `+` Suma:  $2+2 = 4$
- `-` Resta:  $2-1 = 1$
- `*` Multiplicación:  $2*3 = 6$
- `/` División:  $10/2 = 5$
- `//` División entera:  $10//3 = 3$
- `%` Módulo:  $10 \% 3 = 1$

## Operadores relacionales

- `==` Igual
- `!=` Diferente
- `<` Menor que
- `>` Mayor que
- `<=` Menor o igual
- `>=` Mayor o igual

## Condición

Es una expresión lógica que sirve para evaluar si es verdadero o falso, por lo consiguiente si se ejecuta o no.

Ejemplo:

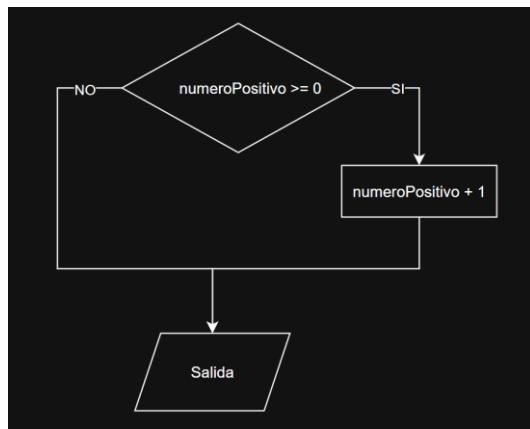
- `numeroPositivo >= 0`

## **Estándar Grafico de estructura Condicional**

Se presenta el diagrama en la página 6 de este glosario.

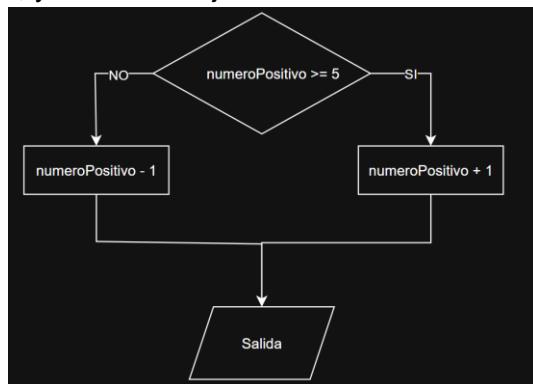
### **Estructura condicional simple**

Solo se ejecuta si la condición es verdadera, de lo contrario no se ejecuta. Usamos IF-THEN.



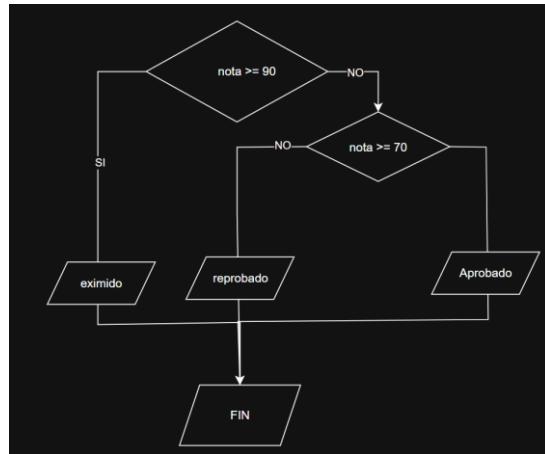
### **Estructura condicional doble**

Se ejecuta tanto si la condición es verdadera o es falsa, es decir que si es verdadera ejecuta una acción, y si es falsa ejecuta otra acción. Usamos IF/ELSE.



### **Estructura condicional Multiple**

Se ejecuta cuando tenemos más de dos posibles resultados sirve para evaluar diferentes condiciones de forma secuencial. Usamos IF-ELIF-ELSE



### Estructura condicional anidada

Contienen una condición dentro de otra, se utiliza cuando se necesita evaluar una condición solo si la primera condición es verdadera.

