

# Programación

**Cristian Pastrana**

## ¿Qué es un programa?

Un **programa** en informática es un conjunto de instrucciones escritas en un **lenguaje de programación** que una computadora puede entender y ejecutar para realizar una tarea específica.

Un **lenguaje de programación** es el **idioma** que usamos para escribir programas que las computadoras puedan entender.

Algunos de los lenguajes mas utilizados en la actualidad son Python, JavaScript, Java, C#, C++, PHP, Go, etc.



## ¿Qué es un programador?

El **programador** es la persona encargada de diseñar, escribir, probar y mantener estos programas. Su actividad fundamental es resolver problemas empleando una computadora como herramienta fundamental.

Para la resolución de un problema el programador debe saber cómo plantear un **algoritmo**.



## Algoritmo

Un **algoritmo** es un conjunto finito de pasos o instrucciones bien definidos que se siguen para resolver un problema o realizar una tarea específica.

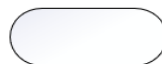
- **Finito:** Tiene un número limitado de pasos.
- **Definido:** Cada paso es claro y no deja lugar a ambigüedades.
- **Entradas:** Puede tener cero o más entradas.
- **Salidas:** Produce al menos una salida.
- **Eficiente:** Debe utilizar los recursos (tiempo y memoria) de forma óptima.



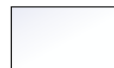
## Diagrama de flujo

Un diagrama de flujo es la representación gráfica de un **algoritmo**. Utiliza símbolos estandarizados para mostrar la secuencia de pasos que se deben seguir para resolver un problema o realizar una tarea.

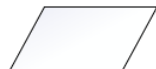
Cada símbolo representa una acción específica.



Inicio y fin del diagrama



Operación



Entrada de datos



Condición



Salida de datos



## Ejemplo de un diagrama de flujo

Problema:

*Calcular el precio total de un producto conociendo su precio y la cantidad de unidades compradas.*

Si hacemos un análisis del problema podemos identificar lo siguiente:

### **Datos conocidos:**

- Precio del producto
- Cantidad de unidades del producto

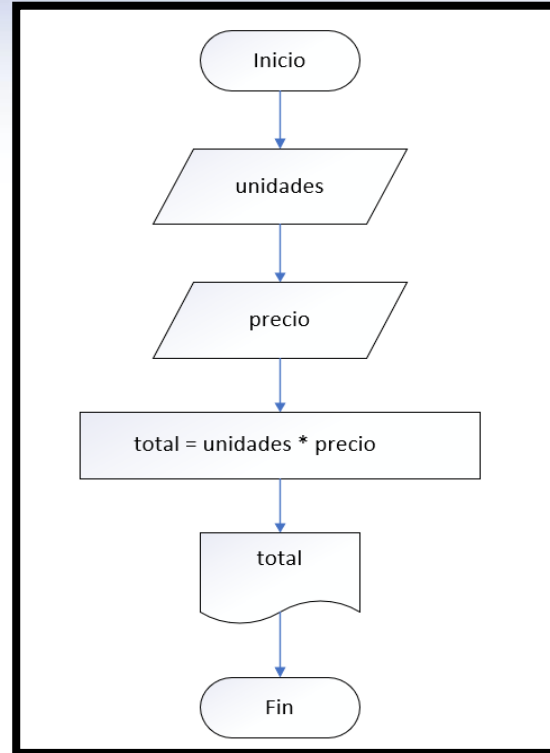
### **Proceso:**

- Calculo del precio total

### **Información resultante:**

- Precio total

Resolución:



## Variables y constantes

Una **variable** es un nombre que se usa para guardar un valor que puede cambiar durante la ejecución del programa.

Ejemplo:

```
nombre = "Juan"  
edad = 25  
edad = 26
```

Una **constante** es un variable donde vamos a guardar un valor que no vamos a cambiar durante la ejecución del programa.

Ejemplo:

```
PI = 3.1416  
TASA_IVA = 0.21  
MAX_USUARIOS = 100  
MENSAJE_BIENVENIDA = "¡Bienvenido al sistema!"
```



## Nombres incorrectos para variables

```
valor inicial  # No debe contener espacios
valor-inicial  # No debe contener - , + , / , *
1valor        # No debe comenzar con un número
print         # print es un nombre reservado (función incorporada)
@valor        # no puede comenzar con símbolos especiales # , @
```

Reglas básicas para nombres de variables válidos:

- Solo pueden contener **letras**, **números** y **guiones bajos** (**\_**)
- No pueden comenzar con un número.
- No pueden usar palabras reservadas del lenguaje.
- Se recomienda usar **nombres descriptivos** y con **letras minúsculas**.



## Ingresar datos por teclado

La **función input()** en Python se utiliza para pedir al usuario que ingrese datos por teclado.

Ejemplo:

```
nombre = input("Hola, ¿Cuál es tu nombre? ")
```

Luego cuando ejecutemos el programa veremos por pantalla:

```
Hola, ¿Cuál es tu nombre?
```

## Mostrar datos por pantalla

La **función print()** en Python se utiliza para mostrar información por pantalla.

Ejemplo:

```
print("Hola, Bienvenido al sistema")
```

Luego cuando ejecutemos el programa veremos por pantalla:

```
Hola, Bienvenido al sistema
```

También podremos mostrar el dato almacenado en una variable.

```
apellido = "Perez"  
print("El valor del apellido es: " + apellido)
```

Luego cuando ejecutemos el programa veremos por pantalla:

```
El valor del apellido es: Perez
```

## Mostrar datos por pantalla

**Importante:** el operador + permite sumar variables entre si, esto funcionara correctamente cuando sumemos valores que sean compatibles. Si sumamos un valor numérico con otro valor numérico por ejemplo (2+3) o (2 + 1.5) serian casos correctos , de la misma forma si sumamos un texto con otro valor de tipo texto ejemplo:

```
("Pedro" + "Perez") dará como resultado "PedroPerez"
```

¿Que sucede si sumo un valor entero con un valor de tipo texto?

```
("la edad es " + 2 ) dará como resultado un error al momento de ejecutar el programa
```

¿Cómo solucionamos este problema?

Lo correcto seria convertir el valor de tipo numérico (entero) a un valor de tipo texto (cadena de texto o string).

La función **str()** permite convertir el tipo de una variable a cadena de texto.

```
("la edad es " + str(2)) dará como resultado "la edad es 2"
```

## Mostrar datos por pantalla

Otra forma de solucionar el anterior inconveniente es utilizar una coma en la función `print()` para mostrar por pantalla más de un valor.

**Ejemplo:**

```
edad = 22  
print("La edad del usuario es : ", edad)
```

Luego cuando ejecutemos el programa veremos por pantalla:

```
La edad del usuario es : 22
```

De esta forma la función `print()` se encarga de mostrar el valor por pantalla sin importar el tipo.

## Tipos de datos básicos

Tipo	Nombre	Ejemplo	Descripción	Función de conversión
int	Entero	10	Números enteros, positivos o negativos	int()
float	Número con decimales	3.14	Números reales con parte decimal	float()
str	Cadena de texto	"Hola"	Texto, puede incluir letras, números o símbolos	str()
bool	Booleano	True, False	Valores lógicos (verdadero o falso)	bool()

## Operadores aritméticos

Operador	Nombre	Descripción	Ejemplo	Resultado
+	Suma	Suma dos valores.	5 + 3	8
-	Resta	Resta el segundo valor al primero.	8 - 2	6
*	Multiplicación	Multiplica dos valores.	6 * 2	12
/	División	Divide el primer valor por el segundo. El resultado es un número decimal.	9 / 2	4.5
//	División entera	Divide y devuelve solo la parte entera del resultado.	9 // 2	4
%	Módulo (residuo)	Devuelve el resto de una división.	9 % 2	1
**	Potenciación (exponente)	Eleva el primer número a la potencia del segundo.	2 ** 3	8

## Operadores relacionales

Operador	Descripción	Ejemplo	Resultado
<b>==</b>	Igual a	<b>5 == 5</b>	<b>True</b>
<b>!=</b>	Distinto de	<b>5 != 5</b>	<b>False</b>
<b>&gt;</b>	Mayor que	<b>7 &gt; 4</b>	<b>True</b>
<b>&lt;</b>	Menor que	<b>8 &lt; 2</b>	<b>False</b>
<b>&gt;=</b>	Mayor o igual que	<b>6 &gt;= 6</b>	<b>True</b>
<b>&lt;=</b>	Menor o igual que	<b>4 &lt;= 5</b>	<b>True</b>