

¿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. Estos pueden ir desde operaciones simples, como una calculadora, hasta complejos sistemas como un navegador web o un videojuego.

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, los principales para el planteo de un diagrama son los siguientes:



Inicio y fin del diagrama



Operación



Entrada de datos



Condición



Salida de datos

Planteo de un problema utilizando un diagrama de flujo

Un ejemplo de un problema a resolver puede ser el siguiente:

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

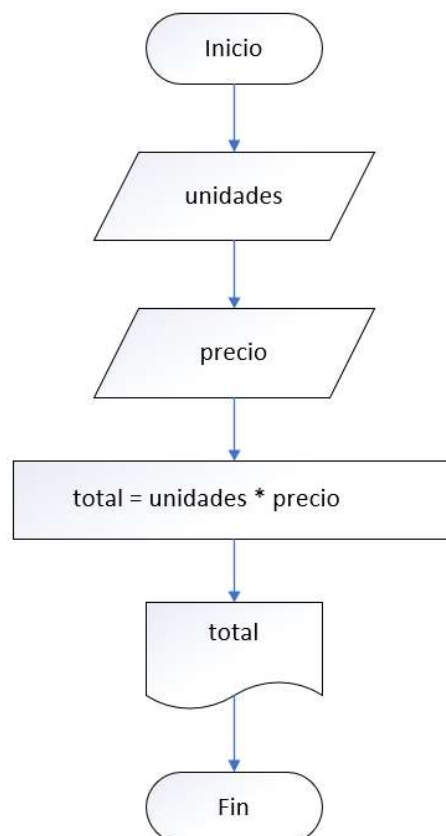
Proceso:

- Calculo del precio total

Información resultante:

- Precio total

Este tipo de análisis resulta sencillo para problemas que no sean complejos, por esta razón es mas efectivo representar los pasos a seguir de nuestro algoritmo mediante un diagrama de flujo.



El **diagrama de flujo** nos da una idea del orden de ejecución de las actividades en el tiempo. Primero cargamos los datos de entrada, luego hacemos las operaciones necesarias y por último mostramos los resultados.

Es importante tener en cuenta como estamos representando el **dato de entrada y de salida**, este valor se conoce como **variable**. Una **variable** consta de un nombre y pertenece a un tipo de dato.

Para el ejemplo dado la **variable** “*unidades*” almacena la cantidad de unidades de un producto, mientras que “*precio*” almacena el precio del producto. Por último, la variable “*total*” almacena el total del precio que debemos cobrar.

En el ejemplo el tipo de dato de las variables corresponde a valores numéricos. Esto tiene sentido ya que no podría multiplicar un valor numérico con algo que no lo sea.

Una variable puede almacenar:

- Valores Enteros (1,2,3,100, etc.)
- Valores Reales (0.50, 10.00, 50.00, etc.)
- Cadenas de caracteres ("Manzanas", "Pedro", "Cafetería", etc.)

Para poder probar nuestra solución propuesta vamos a codificar en **Python** los pasos que definimos en el **diagrama de flujo**.

Python



Es un lenguaje de programación conocido por su **sintaxis simple y clara**, lo que lo hace ideal para personas que están aprendiendo a programar.

Sus principales características son:

- **Lenguaje de programación de alto nivel:** Significa que **la sintaxis se parece más al lenguaje humano** que al lenguaje de la máquina. Por lo cual es más fácil de leer, escribir y entender.
- **Lenguaje interpretado:** Significa que el código se ejecuta línea por línea a través de un “interprete”. **No hace falta compilar todo el programa** antes de probarlo. Esto permite detectar errores rápidamente y probar cambios al instante.
- **Lenguaje de propósito general:** Quiere decir que Python **no está limitado a un solo tipo de tarea**. Con él podremos crear sitios web, automatizar tareas, analizar datos, crear juegos, hasta inteligencia artificial o robótica.

Codificación en Python

Si codificamos la solución propuesta anteriormente podremos crear un programa similar al siguiente:

```
unidades = input("Ingrese las unidades del producto: ")
unidades = int(unidades)
precio = input("Ingrese el precio del producto: ")
precio = int(precio)
total = unidades * precio
print("El precio total del producto es: ")
print(total)
```

Ahora si ejecutamos nuestro programa veremos que se nos solicita el ingreso de las unidades y el precio del producto, y luego se muestra el precio total.

```
Ingrese las unidades del producto: 10
Ingrese el precio del producto: 5
El precio total del producto es:
50
```

El programa cumple su función correctamente, veamos que hace cada línea de nuestro código.

```
unidades = input("Ingrese las unidades del producto: ")
```

Esta línea de código muestra un mensaje en pantalla solicitando al usuario que ingrese las unidades del producto, el valor ingresado se almacena en la variable unidades como una cadena de texto.

```
unidades = int(unidades)
```

Esta línea convierte el valor almacenado en la variable unidades a entero, esto es necesario ya que no podremos realizar operaciones matemáticas con una cadena de texto.

```
precio=input("Ingrese el precio del producto: ")
```

Al igual que en la variable unidades, en esta línea se almacena el valor del precio del producto ingresado por el usuario.

```
precio = int(precio)
```

También convertimos el valor del precio a un entero.

```
total = unidades * precio
```

Calculamos el total multiplicando las unidades por el precio ingresado, el resultado se almacena en la variable total.

```
print("El precio total del producto es: ")
```

Mostramos por pantalla al usuario un mensaje que indica vamos a dar el total.

```
print(total)
```

Por último, mostramos por pantalla el total que habíamos almacenado en la variable total.

En la codificación del programa se utilizaron algunos elementos que son importantes resaltar. Uno de ellos son las funciones `input()` e `int()` como así también el operador `*` que utilizamos para calcular el precio total. Veamos en detalle que significa cada uno de ellos.

Funciones incorporadas en Python

Las funciones nos permiten hacer tareas comunes sin tener que programarlas, por ejemplo, si no existiera la función `input()` tendríamos que escribir mucho más código para poder solicitarle un dato por teclado al usuario. En resumen, sería como una herramienta ya lista para utilizarse.

Ejemplos de funciones incorporadas en Python:

- `input()` → Para **leer datos** desde el teclado.
- `int()` → Para **convertir** un valor a número entero.
- `print()` → Para **mostrar** información en pantalla.
- `len()` → Para obtener la **longitud** de una lista, cadena, etc.
- `type()` → Para ver el **tipo de dato** de una variable.
- `str()` → Para convertir algo a **texto** (string).
- `float()` → Para convertir a número decimal.

Operador aritmético en Python

Un **operador aritmético** es un **símbolo que se usa para realizar operaciones matemáticas** entre valores o variables.

Python tiene varios operadores aritméticos que ya están definidos y listos para usar en cualquier programa.

Ejemplos de operadores aritméticos en Python:

Operador	Nombre	Descripción	Ejemplo	Resultado
+	Suma	Suma dos valores.	5 + 3	8
-	Resta	Resta el segundo valor al primero.	10 - 4	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