



# Introducción



2

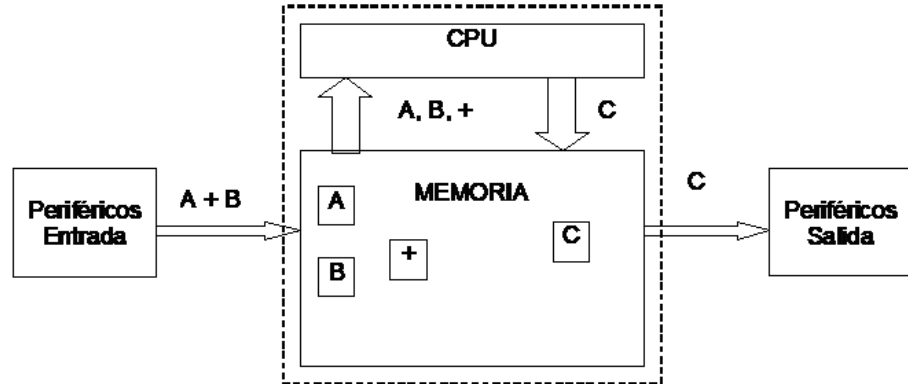
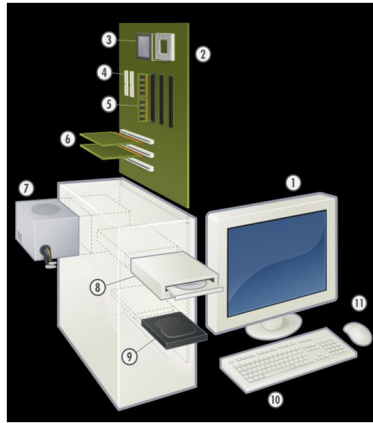
# Contenido

- Introducción
- Python
- Elementos básicos Python
- Ejemplos

# 1. Introducción

## HARDWARE

- Partes físicas: componentes eléctricos, electrónicos, electromecánicos, mecánicos, etc.



# 1. Introducción

## SOFTWARE

- Componentes intangibles: conjunto de programas y procedimientos para que un hardware actúe.

ALTO NIVEL

SOFTWARE DE  
APLICACION

Aplicaciones industriales,  
Aplicaciones ofimáticas,  
Software educativo,  
Software médico,  
Videojuegos.

MEDIO NIVEL

SOFTWARE DE  
PROGRAMACION

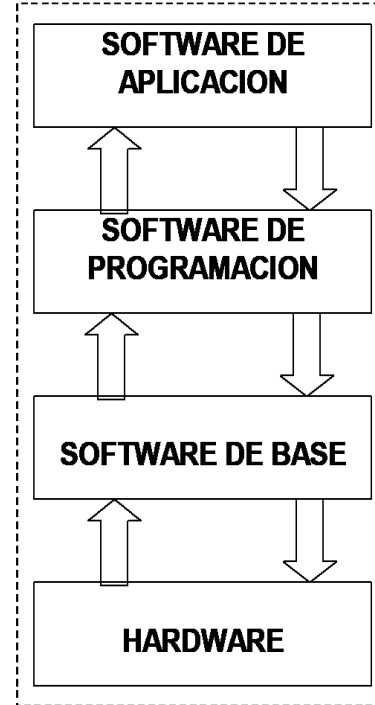
Editores de texto,  
Compiladores,  
Intérpretes,  
Enlazadores,  
Depuradores,

BAJO NIVEL

SOFTWARE DE BASE

Sistemas operativos,  
Controladores de  
dispositivos,  
Servidores.

HARDWARE



# 1. Introducción

## ALGORITMOS Y PROGRAMAS

→ **Algoritmo:** conjunto finito de pasos para resolver un problema.



→ **Programa:** conjunto de instrucciones que se ejecutan en un hardware.

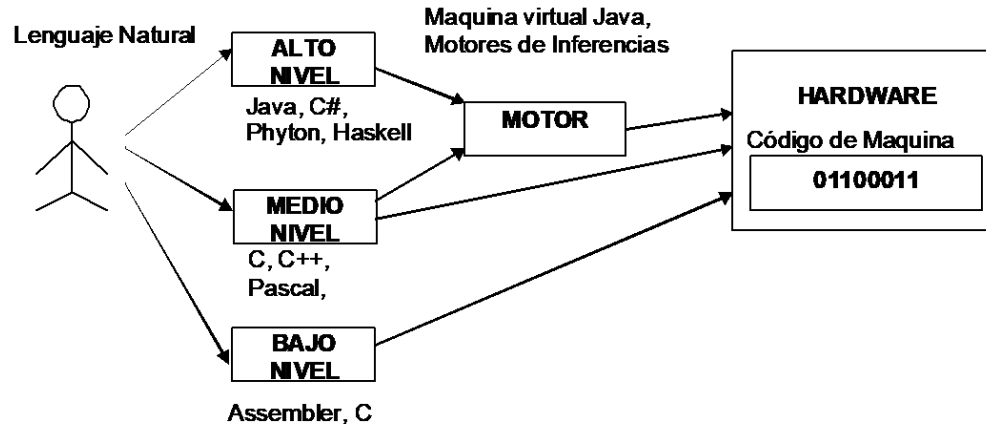


# 1. Introducción

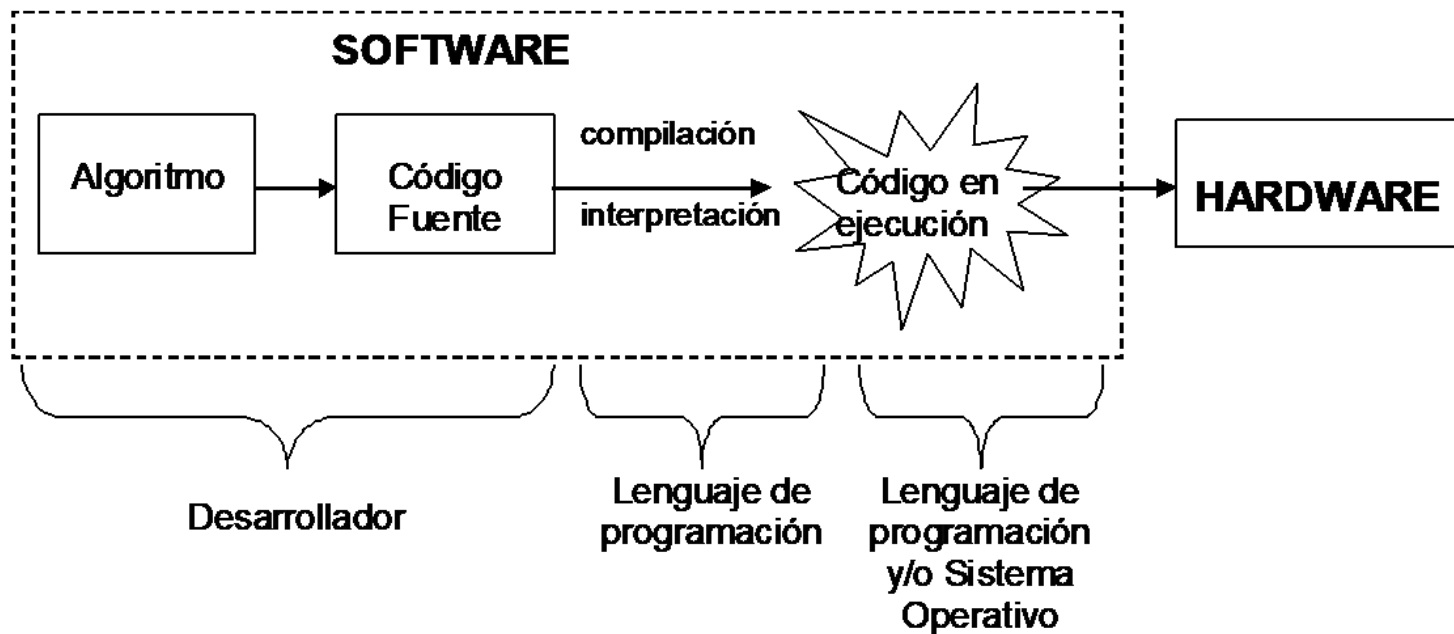
## LENGUAJES DE PROGRAMACIÓN

- Lenguaje formal que le permite al programador escribir un programa para que un hardware actúe.

### Niveles de abstracción de los lenguajes de programación



# 1. Introducción



## 2. Python

### **CARACTERÍSTICAS**

- Lenguaje de programación.
- Interpretado.
- De alto nivel.
- Libre y fuente abierta.
- Multiplataforma (Windows, Unix/Linux, Mac OS X).
- Multiparadigma (estructurado, objetos, funcional).
- Tipado dinámico.



## 2. Python

### ENLACES

- ➔ El intérprete de Python y su extensiva librería estándar se encuentran disponibles libremente en código fuente y forma binaria para la mayoría de plataformas desde la Web de Python, <https://www.python.org/>, y se pueden distribuir libremente.
- ➔ El mismo sitio contiene distribuciones y direcciones a muchos módulos de Python de terceras partes, programas, herramientas y adicionalmente documentación.

## 2. Python

### ENTORNO DE DESARROLLO

- ➔ **Entorno de desarrollo integrado (IDE)** es un sistema de software para el diseño de aplicaciones, que combina un lenguaje de programación y varias herramientas para el desarrollador.
  - ➔ **PC:** Visual Studio Code, PyCharm o cualquier entorno disponible según sistema operativo.
  - ➔ **Web:**
    - ◆ python tutor: <http://www.pythontutor.com/visualize.html#mode=edit>
    - ◆ repl.it: <https://replit.com/languages/python3>
  - ➔ **Android: Pydroid 3 - IDE for Python 3**  
[https://play.google.com/store/apps/details?id=ru.iiec.pydroid3&hl=en\\_IN&gl=US](https://play.google.com/store/apps/details?id=ru.iiec.pydroid3&hl=en_IN&gl=US)

# 3. Elementos básicos Python

## ELEMENTOS BÁSICOS

- Sintaxis.
- Tipos de datos.
- Variables.
- Operadores.
- Funciones de entrada y salida (input y print).

# 3. Elementos básicos Python

## SINTAXIS

- ➔ Conjunto de reglas que deben seguirse al escribir el código fuente:
  - ♦ Comentarios: *#en línea, `"""Múltiples líneas"""`.*
  - ♦ Palabras reservadas: *palabras especiales del lenguaje.*
  - ♦ Identificadores de nombres: *comienzan con una letra, case sensitive, snake\_case, no palabras reservadas.*
  - ♦ PEPs: *Guía de estilos o estándares opcionales (PEP 8).*

# 3. Elementos básicos Python

TIPOS DE DATOS

Tipo (o Clase)	Descripción	Bytes por cada variable	Rango
<b>bool</b>	valores lógicos	1	[False, True]
<b>int</b>	números enteros	dinámico	ilimitado
<b>float</b>	números reales	8	hasta 15 decimales
<b>str</b>	cadenas de caracteres	2 * cantidad de caracteres	Unicode

# 3. Elementos básicos Python

## VARIABLE

- ➔ Grupo de bytes asociado a un *nombre* o *identificador* y a un *tipo de dato*, tiene una *dirección* (ubicación) dentro de la memoria y puede *almacenar información* según el tipo de dato que tenga asociado.

```
#asignación del valor 7 en la variable a  
a = 7
```

<b>a</b>	<b>7</b>
----------	----------

# 3. Elementos básicos Python



## EXPRESIONES

- Fórmula que combina **operandos** (valores constantes o variables) y **operadores** (símbolos de operaciones suma, resta, etc.).
- Ejemplo de expresión aritmética, con asignación:

a = 3

b = 5

c = a + 2 \* b

# 3. Elementos básicos Python

## OPERADORES

➔ Aritméticos

Operador	Significado	Ejemplo de uso
+	suma	$a = b + c$
-	resta	$a = b - c$
*	producto	$a = b * c$
/	división de coma flotante	$a = b / c$
//	división entera	$a = b // c$
%	resto de una división	$a = b \% c$
**	potencia	$a = b ** c$



# 3. Elementos básicos Python

## OPERADORES

### ➔ Relacionales

Opera dor	Significado	Ejempl o	Observaciones
==	igual que	<b>a == b</b>	retorna <i>True</i> si <i>a</i> es igual que <i>b</i> , o <i>False</i> en caso contrario
!=	distinto de	<b>a != b</b>	retorna <i>True</i> si <i>a</i> es distinto de <i>b</i> , o <i>False</i> en caso contrario
>	mayor que	<b>a &gt; b</b>	retorna <i>True</i> si <i>a</i> es mayor que <i>b</i> , o <i>False</i> en caso contrario
<	menor que	<b>a &lt; b</b>	retorna <i>True</i> si <i>a</i> es menor que <i>b</i> , o <i>False</i> en caso contrario
>=	mayor o igual que	<b>a &gt;= b</b>	retorna <i>True</i> si <i>a</i> es mayor o igual que <i>b</i> , o <i>False</i> en caso contrario
<=	menor o igual que	<b>a &lt;= b</b>	retorna <i>True</i> si <i>a</i> es menor o igual que <i>b</i> , o <i>False</i> en caso contrario

# 3. Elementos básicos Python

## OPERADORES

➔ Lógicos

Operador	Significado	Ejemplo
and	conjunción lógica (y)	<code>a == b and y != x</code>
or	disyunción lógica (o)	<code>n == 1 or n == 2</code>
not	negación lógica (no)	<code>not x &gt; 7</code>

### 3. Elementos básicos Python

#### SALIDA ESTANDAR



→ ***print()***: Permite mostrar un mensaje y/o el valor de una variable en la consola estándar de salida.

```
a = 3
```

```
print (a)
```

*Resultado por pantalla: 3*

```
print ('El resultado de a es: ', a)
```

*Resultado por pantalla: El resultado de a es: 3*

```
print ('a')
```

*Resultado por pantalla: a*

# 3. Elementos básicos Python



## ENTRADA ESTÁNDAR

- ➔ ***input()***: Permite obtener en forma de cadena de caracteres, la carga desde el teclado.

```
nom = input('Ingrese su nombre: ')
```

- ➔ Para obtener valores numéricos se debe convertir la función a un tipo numérico, int o float:

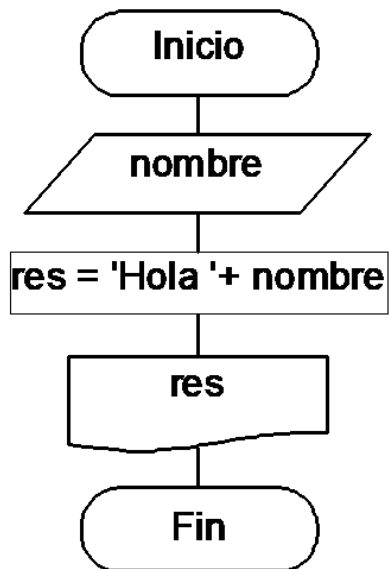
```
n = int(input('Ingrese un valor entero: '))
```

```
x = float(input('Ingrese un valor real: '))
```

## 4. Ejemplos



1. Ingresar un nombre por teclado y mostrarlo por pantalla.



### Pseudocódigo:

1-Cargar **nombre**:  
nombre del  
usuario

2-Armar cadena  
de resultado:  
'Hola' y **nombre**

3-Mostrar el  
resultado **res**

### Código Python:

```
nombre = input("Ingrese  
su nombre: ")  
print("Hola ", nombre)
```

## 4. Ejemplos



2. Dado el valor de los tres lados de un triángulo, calcular el perímetro del triángulo.

Diagrama de Flujo	Script en Python
<pre>graph TD; Inicio([Inicio]) --&gt; Input[/lad1, lad2, lad3/]; Input --&gt; Process[p = lad1 + lad2 + lad3]; Process --&gt; Output[p]; Output --&gt; Fin([Fin]);</pre>	<pre>#ingreso de datos lad1 = float(input('Longitud del primer lado: ')) lad2 = float(input('Longitud del segundo lado: ')) lad3 = float(input('Longitud del tercer lado: '))  #proceso p = lad1 + lad2 + lad3  #visualización de resultado print('El perímetro es:', p)</pre>

## 4. Ejemplos



3. Ingresar el valor de tres temperaturas, calcular y mostrar el promedio de las temperaturas medidas.

### Problema 3) Script en Python

```
#ingreso de datos
t1 = int(input('Temperatura 1: '))
t2 = int(input('Temperatura 2: '))
t3 = int(input('Temperatura 3: '))

#procesos
suma = (t1 + t2 + t3)
promedio = suma / 3

#resultados
print('Promedio: ', round(promedio,2))
```



¡Gracias!

Hasta la próxima

