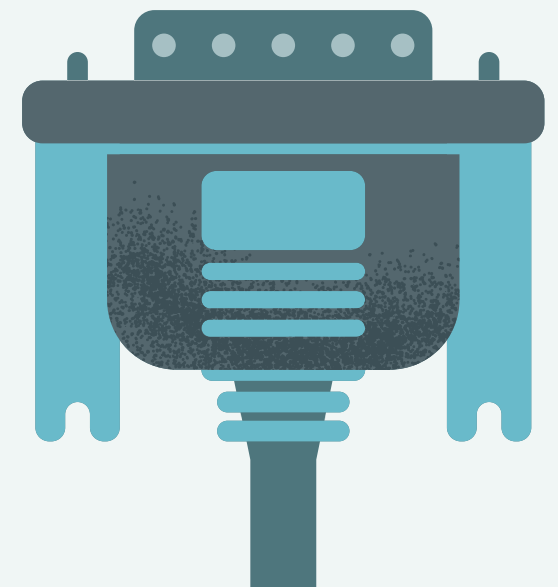
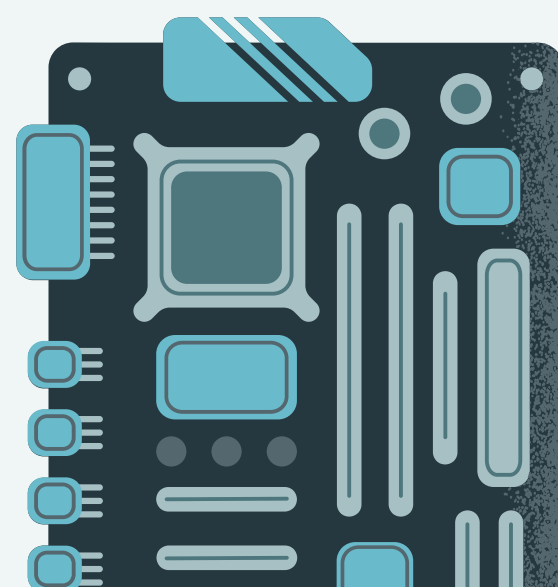
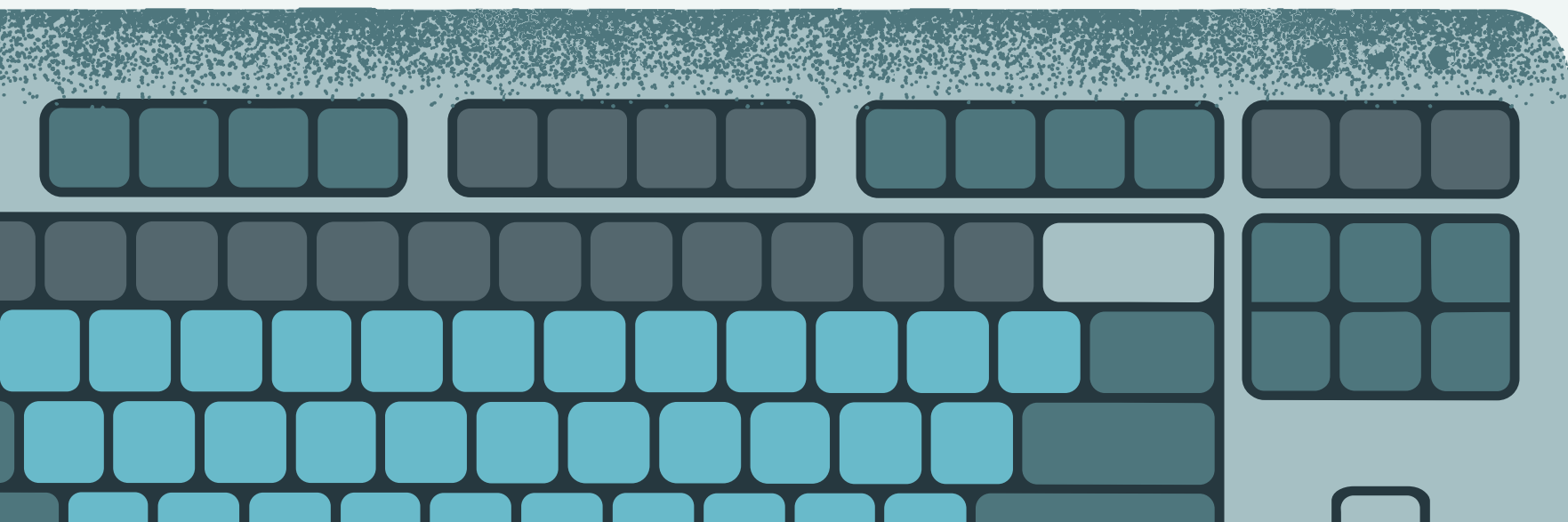
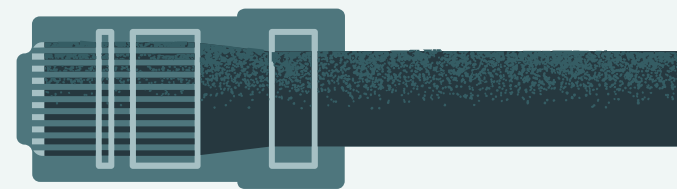
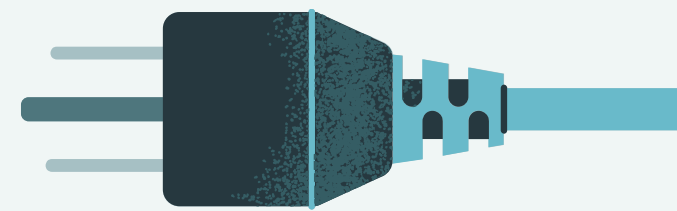
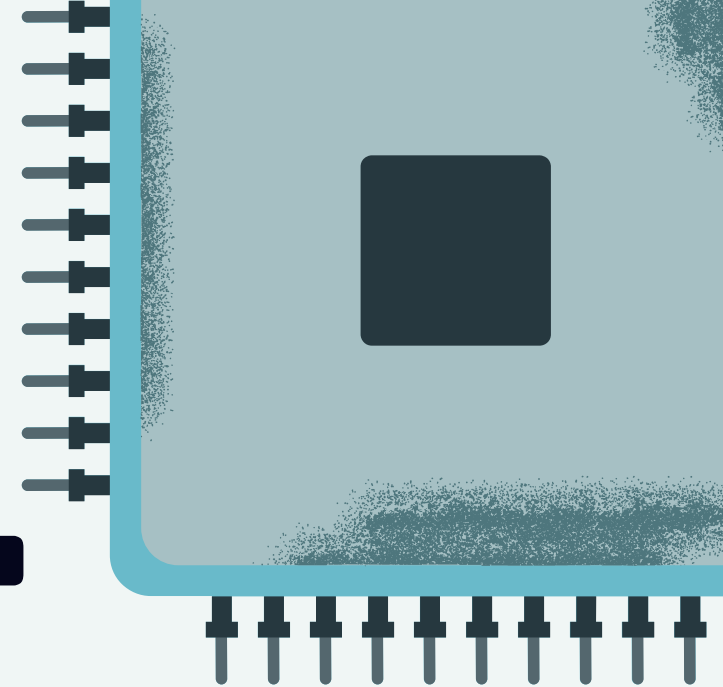


CONOCIENDO EL LENGUAJE PYTHON



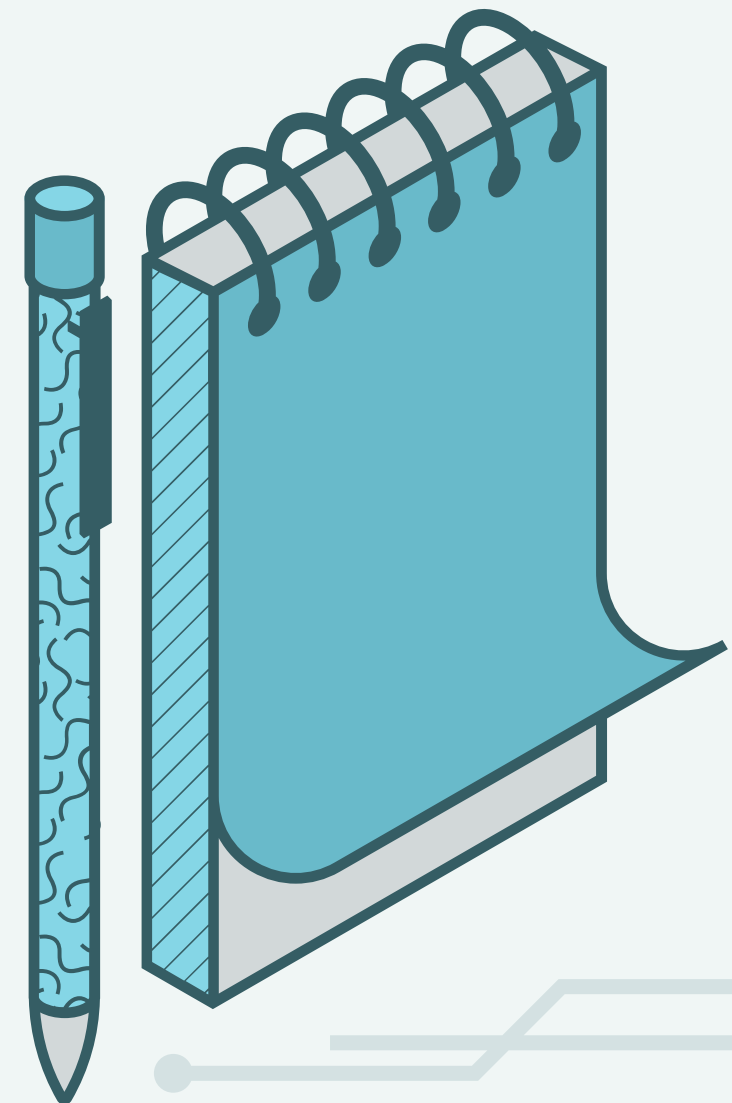
¿QUÉ ES PYTHON?

Python es un lenguaje de programación interpretado, de alto nivel y multipropósito.

Su principal objetivo es ofrecer una sintaxis clara y legible, que permita a los programadores enfocarse en resolver problemas en lugar de preocuparse por la complejidad del código.

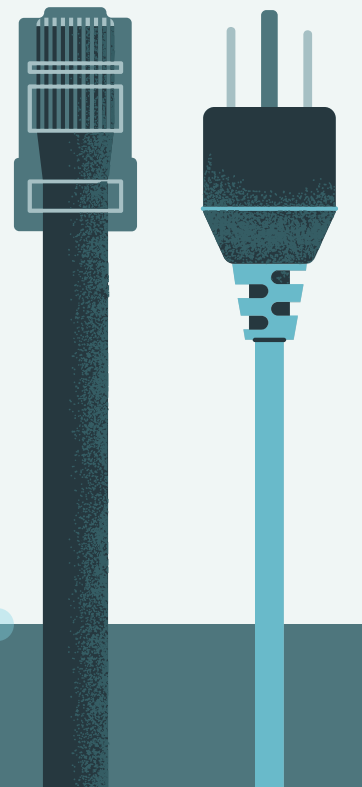
Actualmente, Python es uno de los lenguajes más populares del mundo y se utiliza en:

- Desarrollo web (Django, Flask)
- Ciencia de datos y Machine Learning (NumPy, Pandas, TensorFlow)
- Automatización de tareas
- Desarrollo de videojuegos y aplicaciones
- Internet de las cosas (IoT) con placas como Raspberry Pi

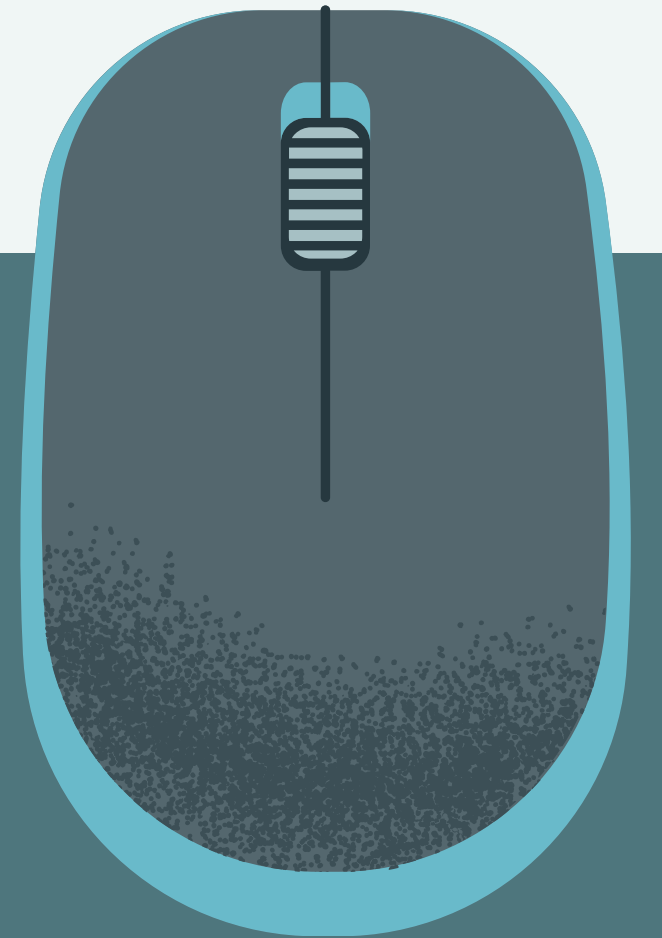


CARACTERÍSTICAS PRINCIPALES

- **Fácil de aprender:** Su sintaxis es similar al lenguaje natural
- **Interpretado:** No se necesita compilar, el código se ejecuta directamente línea por línea.
- **Multiplataforma:** Funciona en Windows, macOS, Linux y más.
- **Orientado a objetos:** Permite organizar el código en clases y objetos, haciendo los programas más estructurados.
- **Versátil:** Se puede usar en casi cualquier área tecnológica.



INSTALACIÓN Y PRIMER PROGRAMA



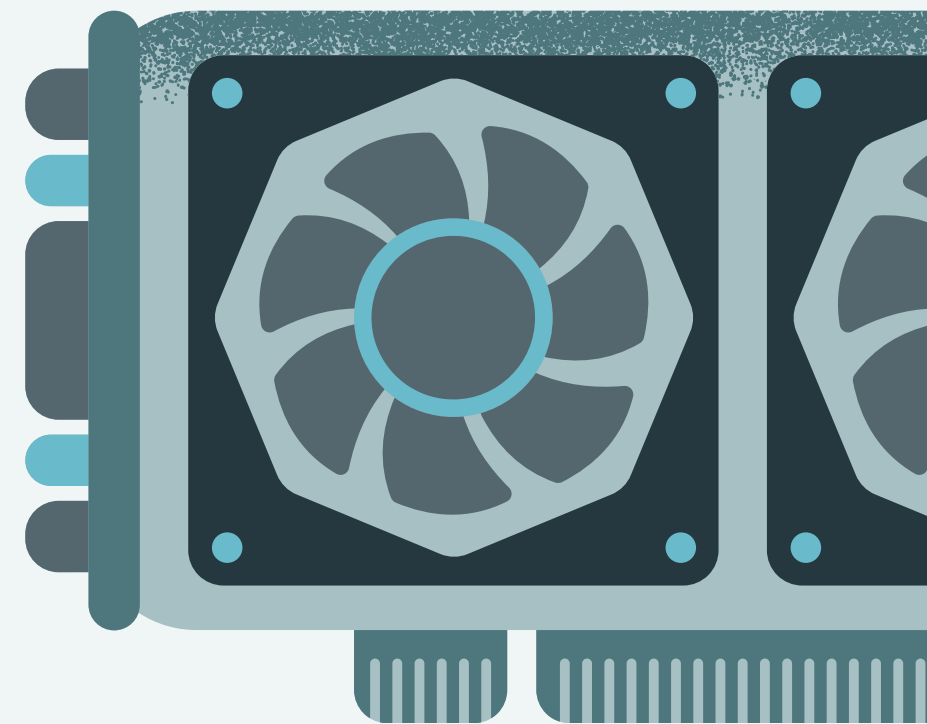
Para empezar a usar Python se puede:

- Descargar desde la página oficial python.org
- Usar entornos como VS Code, PyCharm, Google Colab o el editor que viene integrado: IDLE

Ejemplo del clásico primer programa:
HOLA MUNDO

```
print("Hola, mundo")
```

Este comando imprime un mensaje en pantalla. No necesitas declarar nada antes; solo escribir y ejecutar.



COMENTARIOS EN PYTHON

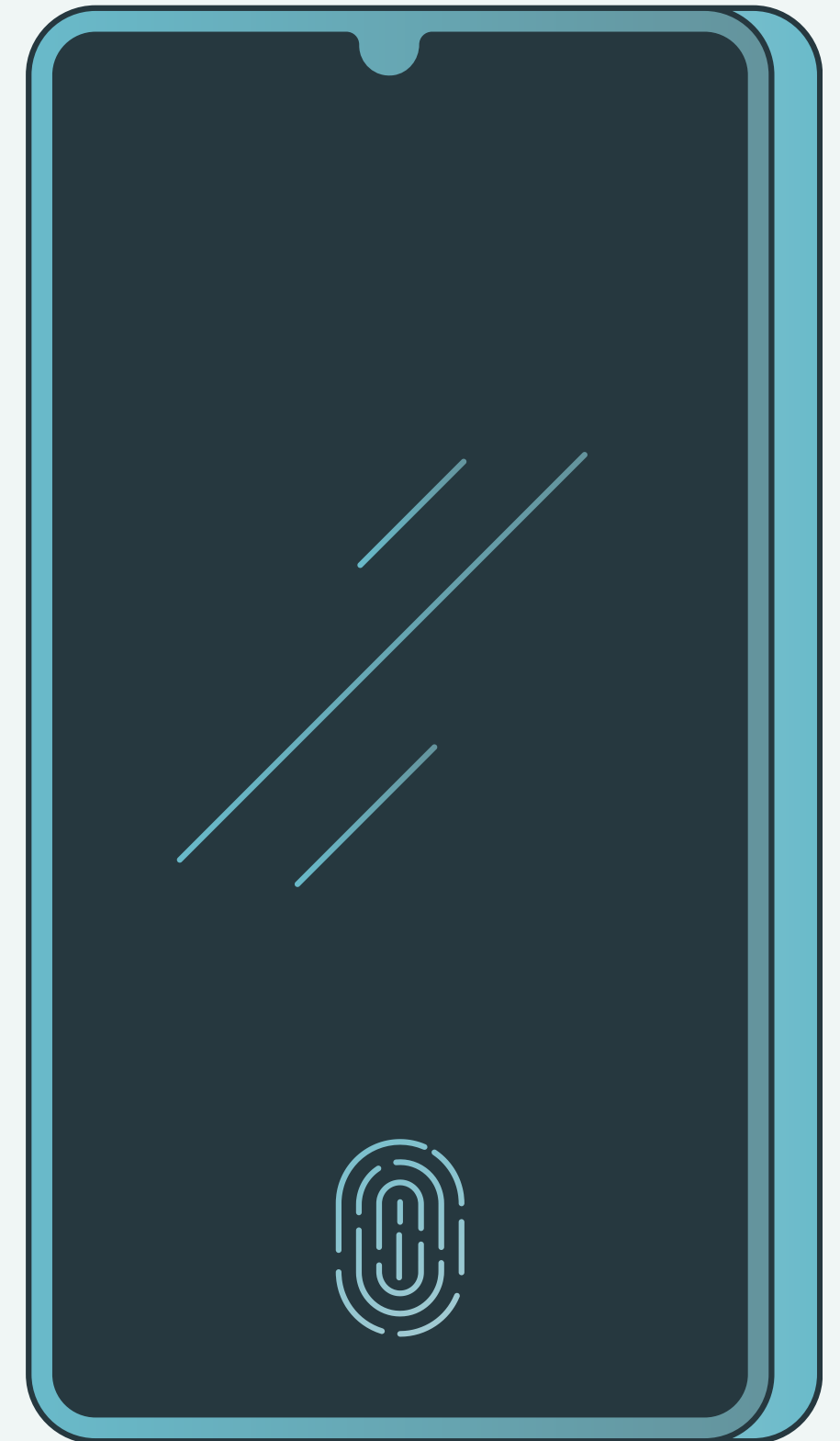
- Los comentarios sirven para explicar el código o dejar notas.
- Python los ignora al ejecutar el programa.
- Existen distintas formas de hacer comentarios

UNA SOLA LÍNEA:

```
# Esto es un comentario
```

VARIAS LÍNEAS:

```
""  
Este es un comentario  
de varias líneas  
""
```



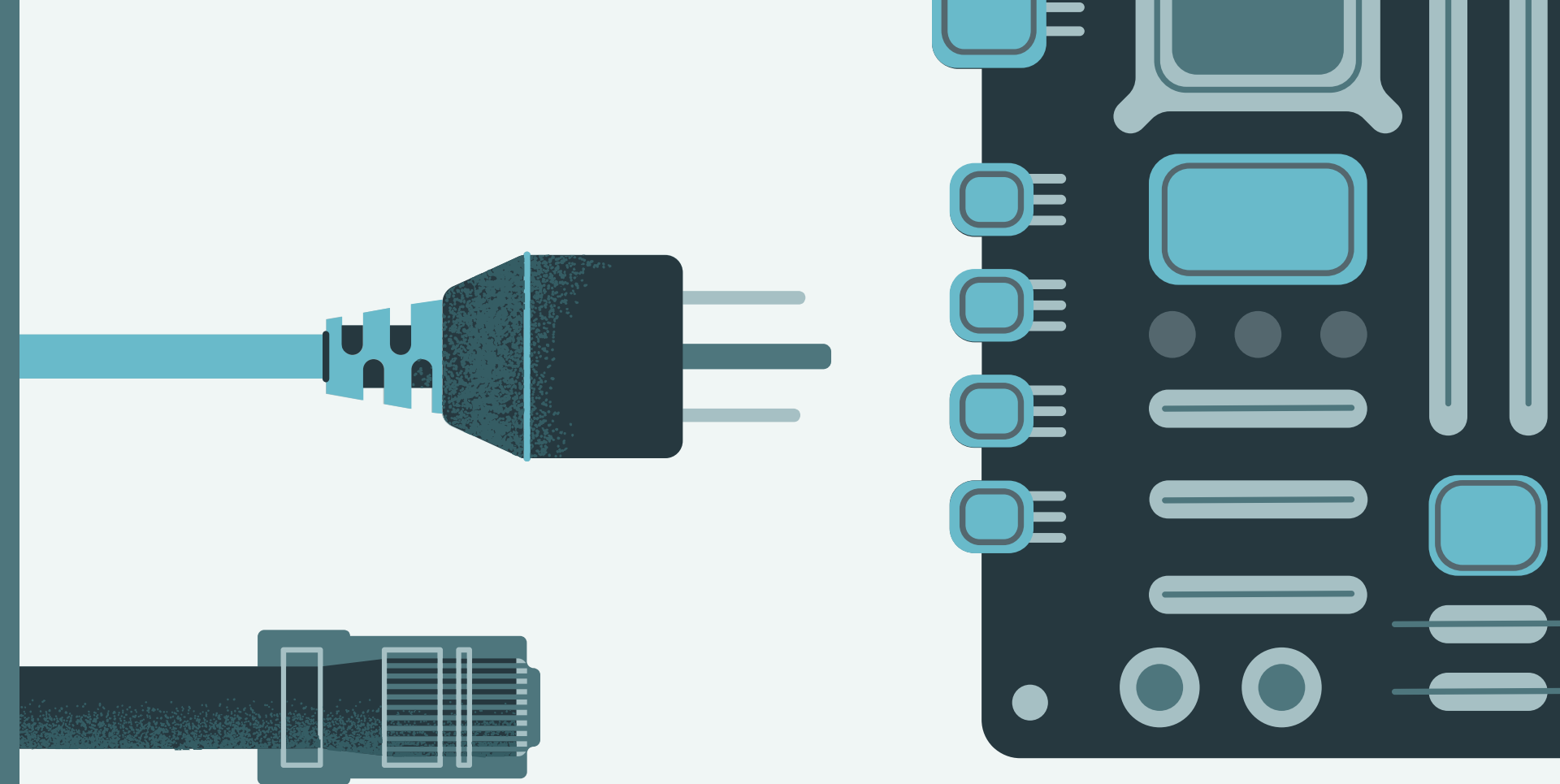
VARIABLES Y TIPOS DE DATOS

Las variables son espacios en memoria donde se guardan valores.

En Python, no es necesario indicar el tipo de dato, el intérprete lo detecta automáticamente.

Principales tipos de datos:

- int → números enteros (5, -3, 100)
- float → números decimales (3.14, 1.75)
- str → texto o cadenas de caracteres (“Hola”)
- bool → valores lógicos (True o False)
- list → listas de elementos



```
nombre = "Luis"      # tipo str (cadena)
edad = 20             # tipo int (entero)
altura = 1.75         # tipo float (decimal)
activo = True         # tipo bool (booleano)
```

BIBLIOTECAS Y MÓDULOS

Python tiene miles de bibliotecas (o módulos) que amplían sus funciones.

Se pueden importar fácilmente con la palabra clave *import*.

```
import math
print(math.sqrt(16)) # calcula la raíz cuadrada
```

EJEMPLOS DE BIBLIOTECAS POPULARES:

- math → operaciones matemáticas
- random → números aleatorios
- datetime → fechas y horas
- os → interacción con el sistema operativo

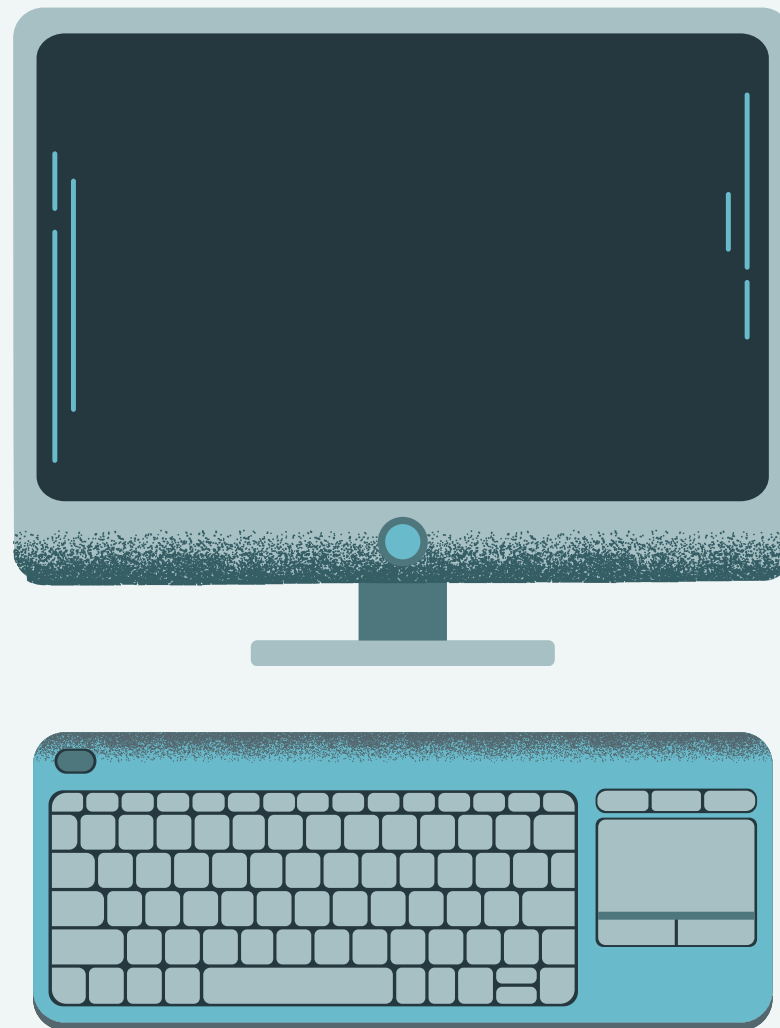


OPERADORES

Los operadores permiten realizar cálculos o comparaciones:

ARITMÉTICOS

+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (resto)
**	Potencia
//	División entera



RELACIONALES

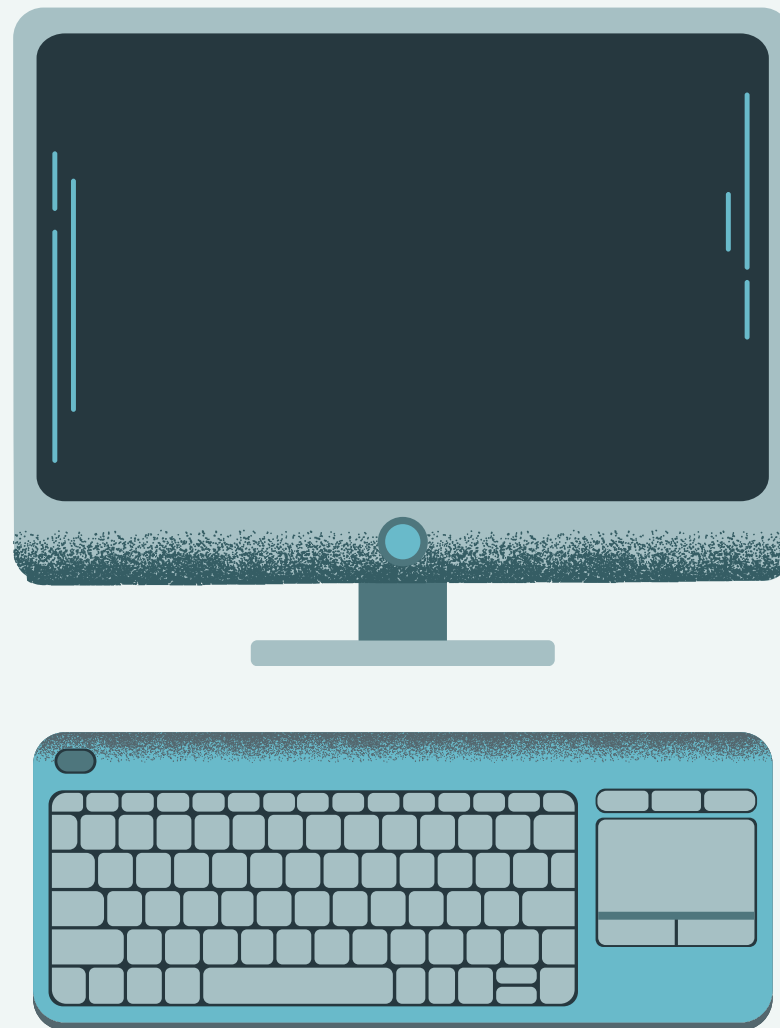
==	Igual a
!=	Diferente de
>	Mayor que
<	Menor que
>=	Mayor o igual
<=	Menor o igual

OPERADORES

Los operadores permiten realizar cálculos o comparaciones:

LÓGICOS

```
and → y  
or → o  
not → no
```

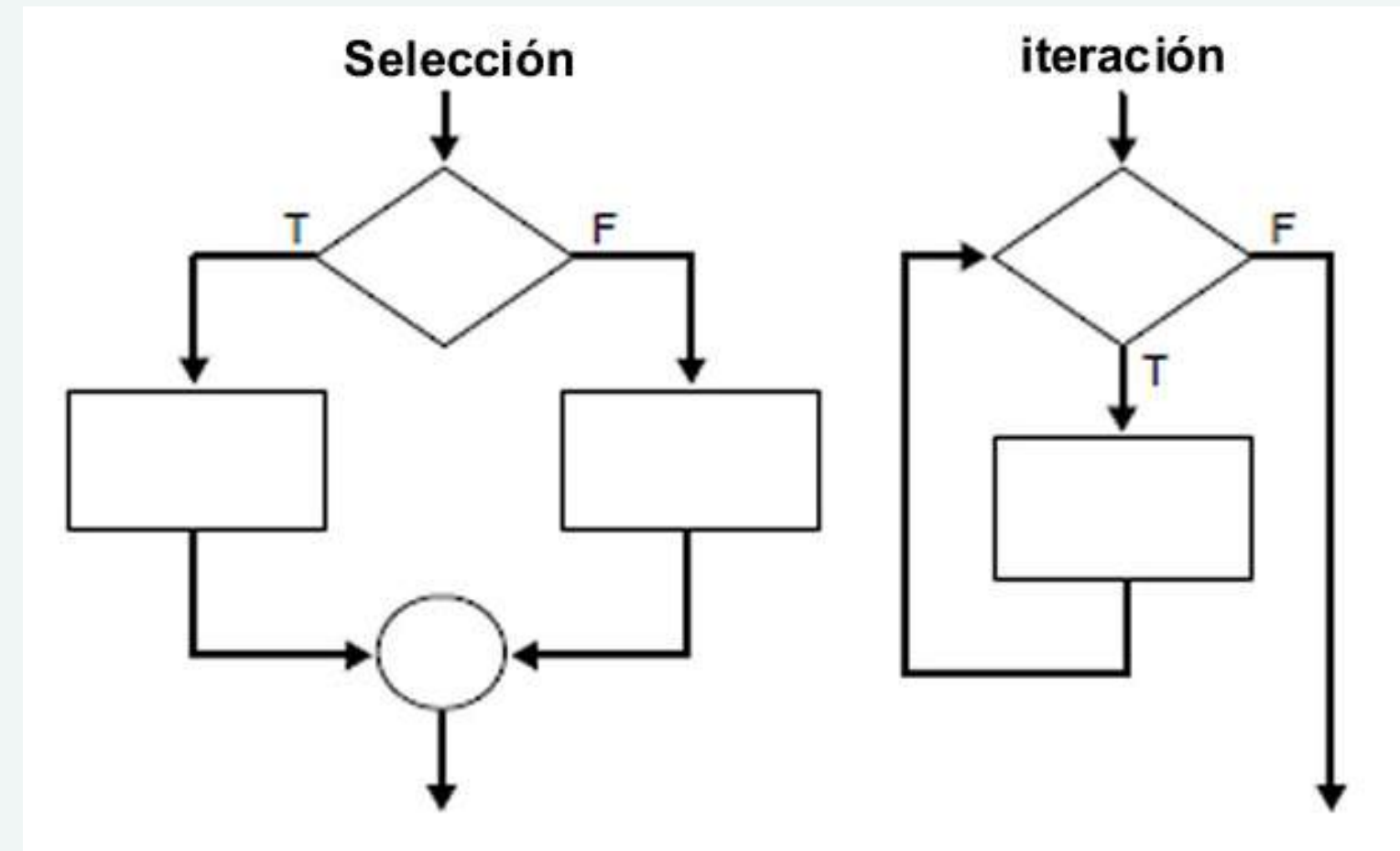
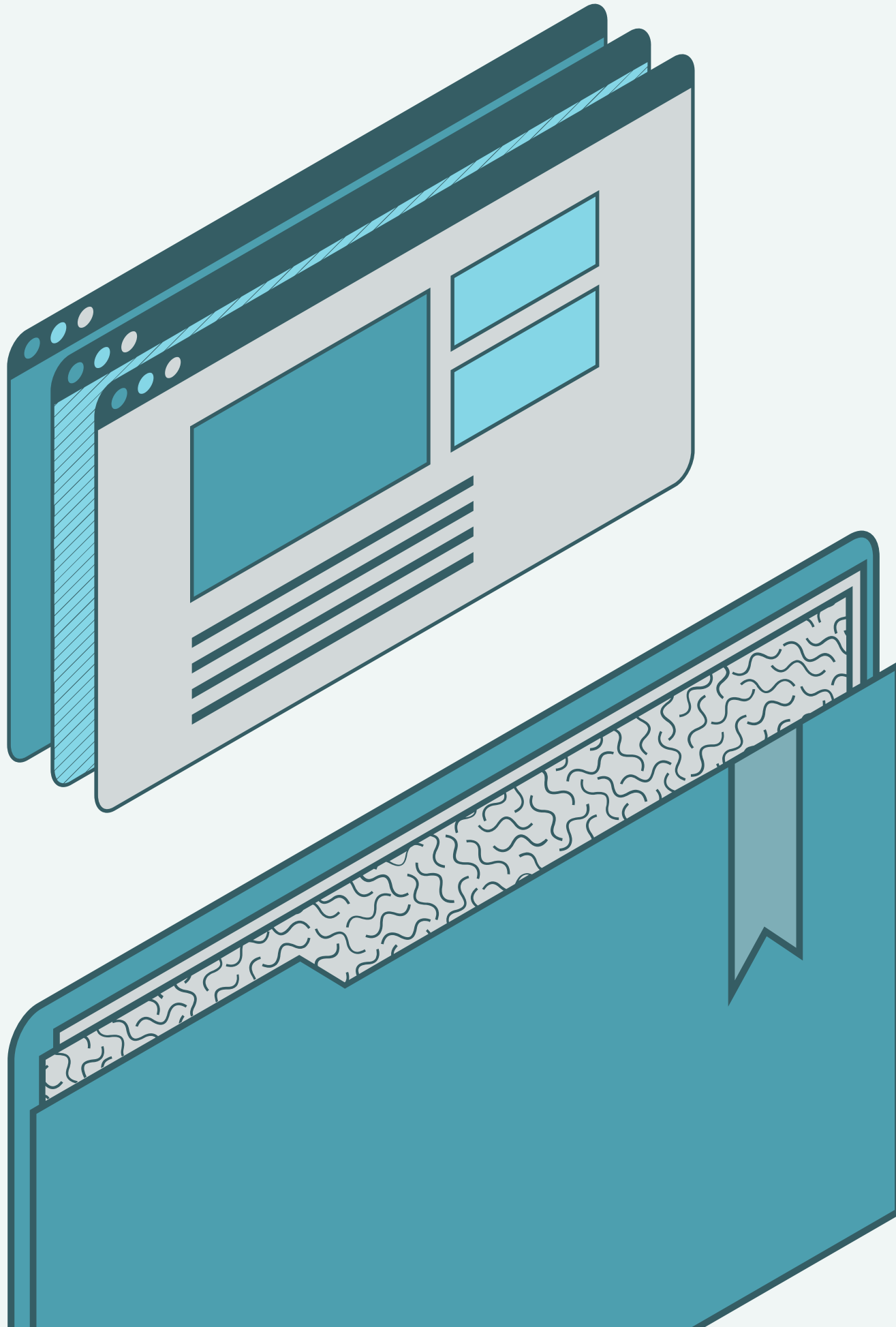


EJEMPLO:

```
x = 10  
y = 5  
print(x > y and y > 0) # True
```

ESTRUCTURAS DE CONTROL

Sirven para tomar decisiones o repetir acciones.



CONDICIONALES (IF, ELIF, ELSE):

IF

Se usa para evaluar una condición.

Si la condición es verdadera (True), el bloque de código dentro del if se ejecuta.

```
if edad >= 18:  
    print("Eres mayor de edad")
```

ELIF

Significa “else if” (si no, entonces si...).

Se utiliza cuando quieres evaluar más de una condición.

Solo se ejecuta si la condición del if anterior fue falsa, pero la del elif es verdadera.

```
elif edad == 17:  
    print("Casi eres mayor de edad")
```

ELSE

Se ejecuta cuando ninguna de las condiciones anteriores fue verdadera.

Funciona como una “opción por defecto”.

```
else:  
    print("Eres menor de edad")
```

BUCLÉS:

FOR

El ciclo for se usa cuando sabemos cuántas veces queremos repetir una acción.

Recorre una secuencia (como una lista, cadena o rango de números).

```
for i in range(5):  
    print(i)
```

WHILE

El ciclo while se usa cuando no sabemos cuántas veces se repetirá el bloque, y queremos que el programa siga ejecutándose mientras se cumpla una condición.

```
contador = 0  
while contador < 5:  
    print(contador)  
    contador += 1
```

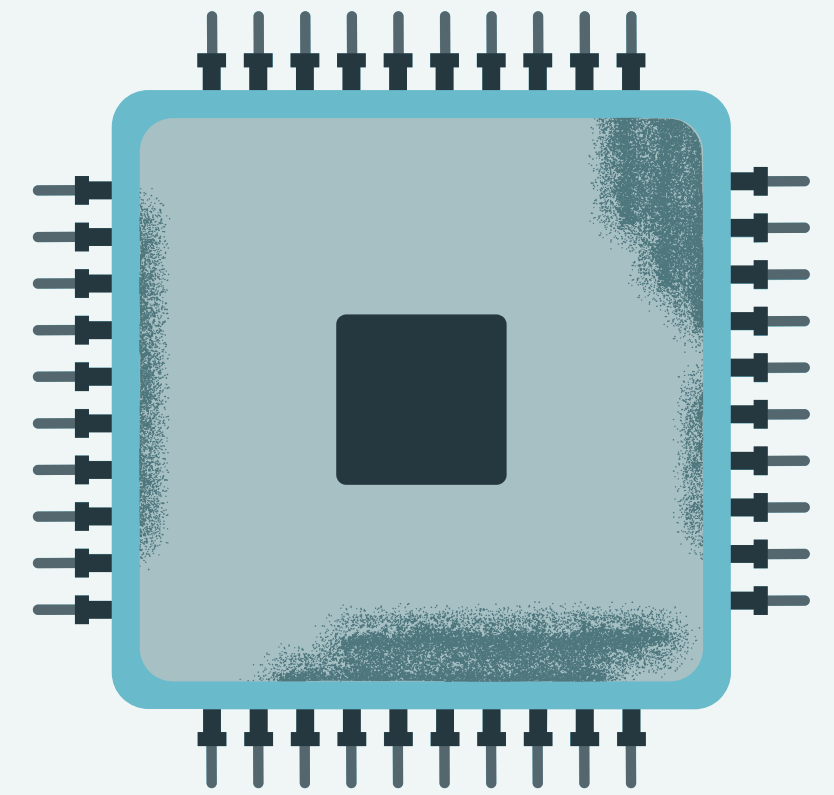
WHILE TRUE

El ciclo while True se utiliza para crear un bucle infinito mientras la condición sea verdadera.

El bucle solo se detiene si usamos la instrucción break cuando se cumple cierta condición.

```
while True:  
    nombre = input("Escribe tu nombre: ")  
    if nombre == "salir":  
        print("Programa terminado.")  
        break  
    print("Hola,", nombre)
```

LISTAS Y DICCIONARIOS



LISTAS:

Almacenan varios elementos y se pueden modificar.

```
frutas = ["manzana", "pera", "uva"]  
print(frutas[0])           # manzana  
frutas.append("naranja") # agrega un elemento
```

DICCIONARIOS:

Guardan información mediante pares clave: valor.

```
persona = {"nombre": "Luis", "edad": 20, "ciudad": "CDMX"}  
print(persona["nombre"]) # Luis
```

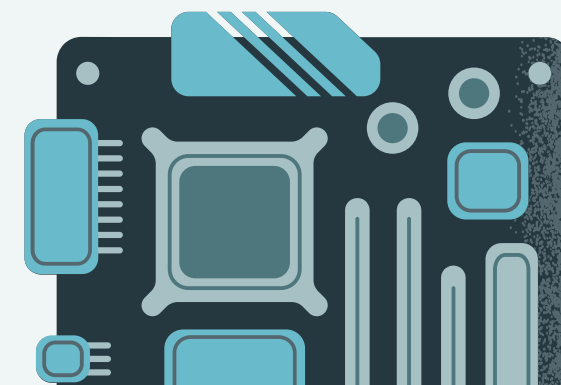
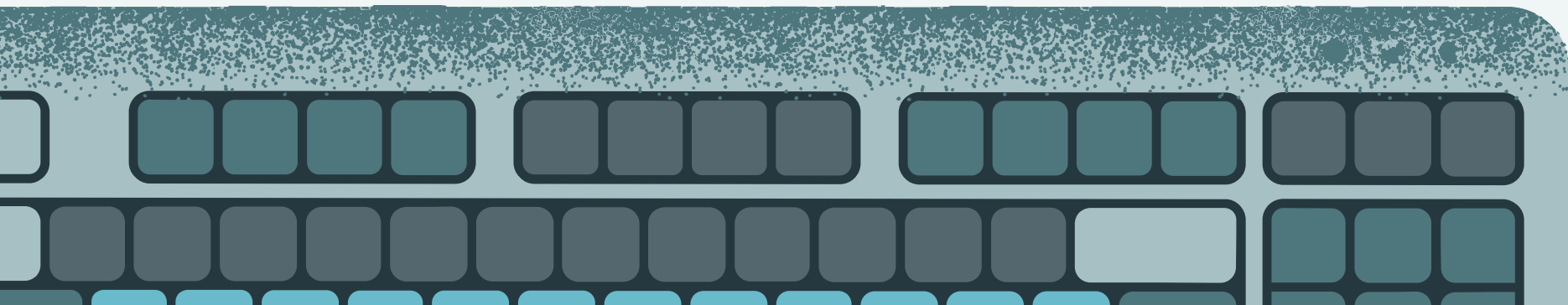
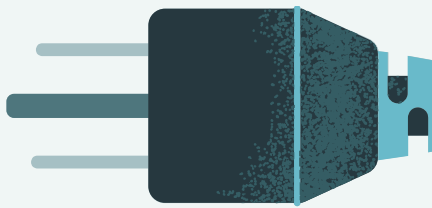
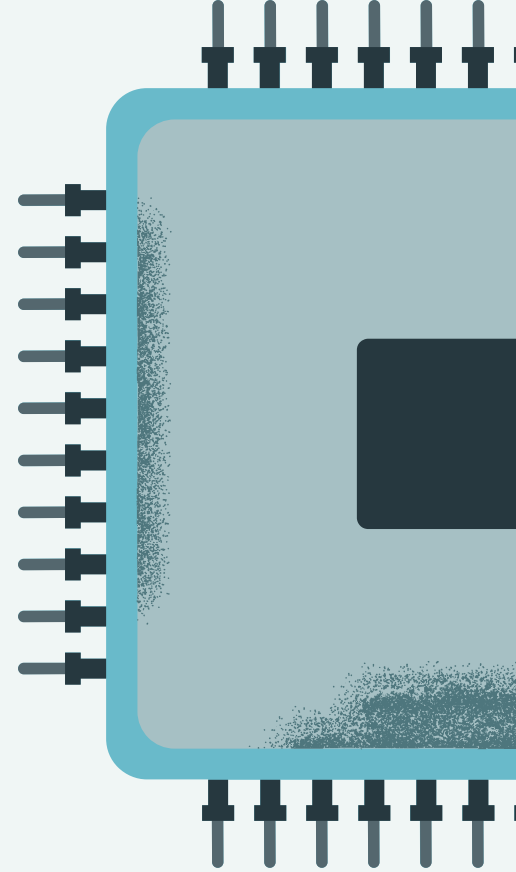
ENTRADA DE DATOS

Permite que el usuario escriba información durante la ejecución del programa.

Se usa la función `input()`:

```
nombre = input("Escribe tu nombre: ")  
print("Hola,", nombre)
```

Todo lo que el usuario ingresa se guarda como texto (str).



FUNCIONES

Las funciones son bloques de código que realizan una tarea específica y pueden reutilizarse varias veces dentro de un programa.

Sirven para organizar y simplificar el código, evitando repetir instrucciones. Se definen con la palabra clave def.

```
def saludo(nombre):  
    print("Hola", nombre)  
  
saludo("Luis")
```

También pueden retornar valores:

```
def sumar(a, b):  
    return a + b  
  
resultado = sumar(5, 3)  
print(resultado) # 8
```



EJEMPLOS DE USOS REALES

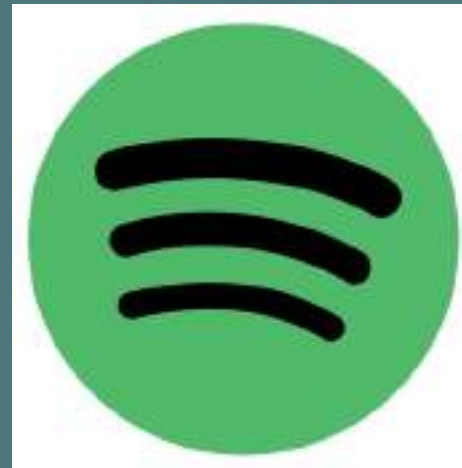
NASA

Utiliza Python en sistemas de control y simulación.



INSTAGRAM Y SPOTIFY

Se utiliza para sus servidores web.



RASPBERRY PI

Usa Python como lenguaje principal para controlar sensores, motores, y proyectos de IoT.



NETFLIX Y YOUTUBE

Se hace uso de python para el análisis de datos.



www.unsitiogenial.es

MUCHAS GRACIAS

