< / > CLUB PROGRAMADOR

COLECCIÓN DE PROGRAMACIÓN / PRIMERA EDICIÓN

100 desafíos de lógica y programación con Python

De los primeros programas a proyectos integradores

```
for desafio in range(1, 101):
    aprender(desafio)
    practicar()
    mejorar()
```

MUESTRA GRATUITA · 10 EJERCICIOS

Fundamentos, índice y primeros desafíos resueltos

La edición completa incluye 100 ejercicios y sus archivos .py.

Créditos editoriales

100 desafíos de lógica y programación con Python

De los primeros programas a proyectos integradores

Autor y editor: Club Programador

Sitio web: https://clubprogramador.com

Contacto: info@clubprogramador.com

Edición: Primera edición · 2026

Formato: Libro digital PDF + archivos fuente .py (ZIP)

Idioma: Español

Derechos y condiciones de uso

© 2026 Club Programador. Todos los derechos reservados. Esta obra y sus materiales originales se ofrecen para el estudio personal del comprador. Queda prohibida su redistribución, reventa o publicación no autorizada, sin perjuicio de las excepciones legales aplicables.

Independencia editorial

Esta publicación es independiente y no está patrocinada, avalada ni afiliada a la Python Software Foundation. Python y los nombres de productos de terceros pertenecen a sus respectivos titulares. Los programas de ejemplo tienen finalidad educativa; su comportamiento puede variar según la versión y el entorno de ejecución.

Erratas y correcciones

Para informar una errata, escribí a info@clubprogramador.com indicando el número de ejercicio, la versión de Python y una descripción del problema. Las correcciones editoriales que se publiquen podrán consultarse en clubprogramador.com. Esto no implica un compromiso de actualizaciones gratuitas ilimitadas.

Agradecimientos

Gracias a la comunidad de Club Programador por compartir sus preguntas, sus soluciones y distintas maneras de resolver un mismo problema. Este libro está dedicado a quienes aprenden practicando y comparten lo que descubren para que otros también puedan aprender.

Antes de empezar

Este libro está pensado para quienes comienzan a programar y para quienes desean dar el salto del pseudocódigo a Python. No es necesario tener experiencia previa: vas a conocer los conceptos esenciales, practicar con problemas progresivos y llegar a proyectos que combinan múltiples técnicas.

Qué incluye esta edición

- Fundamentos de programación, algoritmos y primeros pasos con Python.
- Guía de instrucciones, tipos de datos, estructuras de control y operadores.
- 100 ejercicios numerados, con soluciones y material explicativo.
- Archivos .py individuales para ejecutar y modificar los programas.
- Proyectos integradores con funciones, objetos y persistencia de datos.

Cómo aprovechar cada ejercicio

01. Leé el enunciado e identificá entradas, salidas y restricciones.
02. Intentá escribir una solución propia antes de consultar la propuesta.
03. Compará tu código con el ejemplo, línea por línea.
04. Reproducí los casos de prueba, incluidos datos inválidos y valores límite.
05. Realizá el desafío adicional para reforzar lo aprendido.

> **La estructura de aprendizaje**
>
> Primero comprendé el problema; luego ejecutá y verificá la solución. Al avanzar, incorporá nuevas técnicas de Python y aplicalas para crear programas cada vez más completos.

MUESTRA GRATUITA
Los 100 archivos .py y el apéndice se incluyen en la edición completa.

Archivos fuente y recursos

El libro incluye el archivo Club_Programador_Python_Ejercicios.zip con 100 programas. Cada carpeta, de Ejercicio 01 a Ejercicio 100, contiene el archivo .py correspondiente. Conservá los archivos de ejemplo junto al programa que los utiliza.

Cómo utilizar los archivos .py

01. Descargá e instalá Python 3 desde su sitio oficial.

02. Descomprimí el archivo ZIP en una carpeta de tu computadora.

03. Abrí el archivo deseado con un editor de código, por ejemplo IDLE o Visual Studio Code.

04. Abrí una terminal en Ejercicio 01 y ejecutá python ejercicio_01.py.

05. Probá las entradas propuestas y modificá una copia para resolver el desafío adicional.

Python: fuentes oficiales y recursos

Sitio oficial de Python
https://www.python.org/

Descarga de Python
https://www.python.org/downloads/

Documentación de Python 3 en español
https://docs.python.org/es/3/

Club Programador: recursos y tutoriales
https://clubprogramador.com/python/

Compatibilidad y convenciones

Probamos los 100 ejercicios con Python 3.12 en Windows. Usan la biblioteca estándar. Usá punto como separador decimal y conservá la sangría del código. Si tu sistema usa python3, ejecutá python3 ejercicio_01.py desde la misma carpeta.

Archivos de prueba incluidos

El ejercicio 81 incluye texto.txt y el 83, ventas.csv. Ejecutalos desde sus carpetas. En el ejercicio 85, ingresá hola.txt cuando el programa pida el nombre del archivo. Los ejercicios 84 y 91 guardan cambios en sus archivos JSON. Trabajá sobre una copia si querés conservar los datos iniciales.

MUESTRA GRATUITA
Los 100 archivos .py y el apéndice se incluyen en la edición completa.

ÍNDICE DE CONTENIDOS

Módulos 01 y 02

MÓDULO 01 · Fundamentos y validaciones

01 Control de acceso por edad
02 Convertidor de temperatura
03 Promedio de tres notas
04 Precio con descuento limitado
05 Conversión de segundos
06 Calculadora de consumo de combustible
07 Área y perímetro de un terreno
08 Presupuesto de materiales
09 Conversor de monedas con tasa fija
10 Cálculo de IMC
11 Reparto de gastos entre personas
12 Calculadora de propinas
13 Descomposición de un importe
14 Conversión de unidades de almacenamiento
15 Cálculo de intereses simples

MÓDULO 02 · Decisiones y reglas de negocio

16 Clasificación de calificaciones
17 Control de stock mínimo
18 Tarifa de estacionamiento
19 Costo de envío por zona
20 Cálculo de tarifa eléctrica
21 Clasificador de triángulos
22 Año bisiesto
23 Validación de fecha
24 Comparador de ofertas
25 Descuento por volumen
26 Comisión por ventas
27 Tarifa de transporte
28 Precio de entradas según edad
29 Control de presupuesto
30 Simulador de promociones combinadas

ÍNDICE DE CONTENIDOS

Módulos 03 y 04

MÓDULO 03 · Ciclos y algoritmos numéricos

31 Tabla de multiplicar
32 Suma de una secuencia
33 Contar positivos, negativos y ceros
34 Promedio con cantidad variable
35 Caja registradora con centinela
36 Factorial de un número
37 Sucesión de Fibonacci
38 Divisores de un entero
39 Detección de números primos
40 Primos en un intervalo
41 Máximo común divisor
42 Mínimo común múltiplo
43 Suma de dígitos
44 Número capicúa
45 Menú de operaciones repetibles

MÓDULO 04 · Listas, tuplas y algoritmos

46 Estadísticas de una lista
47 Mayor y menor sin max ni min
48 Inversión de una lista
49 Búsqueda secuencial
50 Búsqueda binaria
51 Ordenamiento burbuja
52 Ordenamiento por selección
53 Ordenamiento por inserción
54 Eliminar duplicados conservando el orden
55 Intercalar dos listas
56 Rotación de una lista
57 Segundo mayor distinto
58 Frecuencia de calificaciones
59 Agrupar valores pares e impares
60 Ranking de resultados

ÍNDICE DE CONTENIDOS

Módulos 05 y 06

MÓDULO 05 · Cadenas, diccionarios y conjuntos

61 Contar vocales y consonantes
62 Palíndromo de texto
63 Analizar palabras de una frase
64 Generar iniciales de nombres
65 Frecuencia de palabras
66 Agenda de contactos
67 Inventario de productos
68 Registro de alumnos por matrícula
69 Comparar conjuntos de participantes
70 Matriz de ventas por sucursal

MÓDULO 06 · Funciones y modularidad

71 Funciones de conversión
72 Calculadora modular
73 Validación reutilizable de enteros
74 Validación reutilizable de decimales
75 Cálculo de cuotas simples
76 Resumen de estadísticas con funciones
77 Generador de reportes de ventas
78 Gestión de tareas con funciones
79 Fibonacci mediante recursividad
80 Separar lógica de negocio y consola

ÍNDICE DE CONTENIDOS

Módulos 07 y 08

MÓDULO 07 · Archivos, excepciones y objetos

81 Lectura de un archivo de texto
82 Guardar notas en CSV
83 Procesar un CSV de ventas
84 Persistir una agenda en JSON
85 Gestionar errores de archivos
86 Clase Producto
87 Clase Cuenta de puntos
88 Clase Alumno y calificaciones
89 Composición: pedido y artículos
90 Pruebas unitarias de una calculadora

MÓDULO 08 · Proyectos integradores

91 Gestor de tareas con persistencia JSON
92 Sistema de biblioteca personal
93 Control de inventario de un comercio
94 Registro de gastos personales
95 Gestión de notas de un curso
96 Sistema de turnos por consola
97 Batalla naval en modo texto
98 Juego de preguntas y respuestas
99 Sistema de reservas de cine
100 Sistema de ventas con catálogo y reportes

APÉNDICE Consulta rápida de Python

Primeros pasos con Python

¿Qué es programar?

Programar es construir una secuencia de instrucciones que resuelve un problema. Python permite expresar esa lógica con una sintaxis clara y ejecutarla directamente.

Del pseudocódigo a Python

En PSeInt se utiliza Escribir y Leer; en Python, print() e input(). Las asignaciones se realizan con =. La indentación indica qué instrucciones pertenecen a un bloque.

Variables y tipos

int representa enteros; float, números decimales; str, texto; bool, valores True o False. input() devuelve texto: utilizá int() o float() cuando necesites un número.

Operadores esenciales

+ - * / realizan operaciones aritméticas. // es división entera, % es resto y ** es potencia. Para comparar se usa ==, !=, <, >, <= y >=. En condiciones lógicas se usan and, or y not.

Estructuras de control

if, elif y else permiten tomar decisiones. for recorre elementos de una secuencia; while repite mientras la condición sea verdadera. try/except permite reaccionar ante errores de ejecución.

> **Archivos y ejecución**
>
> Abrí una terminal en la carpeta Ejercicio 01 y ejecutá python ejercicio_01.py
> (o python3 ejercicio_01.py, según tu instalación). Usá punto como separador decimal.

Indentación en Python

La indentación (o sangría) es el espacio al comienzo de una línea. Python la usa para decidir qué instrucciones pertenecen a un bloque. Después de los dos puntos de un if, for, while o def, escribí el bloque en las líneas siguientes, con sangría.

```
01   edad = 16
02   if edad >= 18:
03       print("Acceso permitido")
04
05   print("Fin de la verificación")
```

Con edad = 16, el programa solo muestra «Fin de la verificación». Esa línea vuelve al nivel del if y se ejecuta siempre. Si le agregás cuatro espacios, queda dentro del bloque y se ejecuta únicamente cuando la condición es verdadera.

Cómo evitar errores de sangría

Usá cuatro espacios por nivel y alineá las instrucciones de un mismo bloque.
Configurá tu editor para insertar espacios y evitá mezclarlos con tabulaciones.
Si falta la sangría de un bloque, Python muestra un error (IndentationError).

Módulo 01: fundamentos y validaciones

Ejercicios 01-15. Aprendé a capturar datos, validar entradas, transformar valores y presentar resultados. Este módulo contiene quince desafíos desarrollados y probados.

Objetivos de aprendizaje

- Leer y convertir datos de la consola.
- Aplicar comparaciones y cálculos básicos.
- Comprender acumuladores, decisiones y repeticiones.
- Detectar entradas fuera del dominio del problema.
- Comparar resultados con pruebas de ejecución.

> **Criterio de calidad**
>
> Las soluciones se probaron con Python. Las consignas de cada ejercicio indican qué tipos de entrada se esperan. Más adelante se amplían las validaciones con excepciones y funciones reutilizables.

MÓDULO 01 · EJERCICIO 001

01. Control de acceso por edad

El problema

Solicitá una edad entera entre 0 y 120. Repetí la lectura si el dato no es válido. Informá si la persona es mayor de edad (18 años o más).

Solución en Python

```python
while True:
    try:
        edad = int(input("Edad (0 a 120): "))
        if 0 <= edad <= 120:
            break
    except ValueError:
        pass
    print("Edad invalida.")

if edad >= 18:
    print("Mayor de edad")
else:
    print("Menor de edad")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
edad = -3	Se rechaza: fuera del intervalo.
edad = 18	Entrada válida; se imprime Mayor de edad.

Casos de prueba

-1, 0, 17, 18, 120, 121 y texto no numérico.

> **Error frecuente**
>
> Confundir una edad válida con la condición de mayoría de edad.

> **Desafío adicional**
>
> Agregá una categoría para las personas de 65 años o más.

02. Convertidor de temperatura

El problema

Leé una temperatura en grados Celsius igual o superior a -273,15. Mostrá su equivalencia en Fahrenheit y Kelvin con dos decimales. Para este ejercicio, suponé entradas numéricas finitas.

Solución en Python

```python
celsius = float(input("Temperatura en Celsius: "))
while celsius < -273.15:
    print("Temperatura invalida.")
    celsius = float(input("Temperatura en Celsius: "))

fahrenheit = celsius * 9 / 5 + 32
kelvin = celsius + 273.15

print(f"Fahrenheit: {fahrenheit:.2f}")
print(f"Kelvin: {kelvin:.2f}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
celsius = 25	Fahrenheit = 25 × 9 / 5 + 32 = 77.
kelvin	25 + 273,15 = 298,15.

Casos de prueba

0 → 32,00 °F y 273,15 K; -273,15 → 0,00 K.

Error frecuente

Aplicar el desplazamiento de 32 antes de multiplicar por 9/5.

Desafío adicional

Permití convertir en sentido inverso, de Fahrenheit a Celsius.

03. Promedio de tres notas

El problema

Solicitá tres calificaciones numéricas entre 0 y 10. Repetí cada lectura hasta que sea válida. Calculá el promedio y mostrá si el estudiante aprobó (promedio mayor o igual a 6). Las entradas son numéricas finitas.

Solución en Python

```python
suma = 0.0
for i in range(1, 4):
    nota = float(input(f"Nota {i}: "))
    while not 0 <= nota <= 10:
        nota = float(input("Nota invalida. Reingrese: "))
    suma += nota

promedio = suma / 3
print(f"Promedio: {promedio:.2f}")
if promedio >= 6:
    print("Aprobado")
else:
    print("Desaprobado")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
notas: 4, 6, 8	El acumulador vale 4, luego 10 y finalmente 18.
promedio = 18 / 3	Resultado 6; se imprime Aprobado.

Casos de prueba

4, 6, 8 → 6,00; 5, 5, 5 → 5,00; 0, 10, 8 → 6,00.

Error frecuente

Dividir dentro del ciclo en vez de dividir la suma final.

Desafío adicional

Solicitá una cantidad variable de notas en lugar de exactamente tres.

04. Precio con descuento limitado

El problema

Leé un precio positivo y un descuento porcentual entre 0 y 50. Validá ambos valores numéricos finitos. Calculá el descuento y el precio final con dos decimales.

Solución en Python

```python
from math import isfinite

precio = float(input("Precio: "))
while not isfinite(precio) or precio <= 0:
    precio = float(input("Precio invalido. Reingrese: "))

porcentaje = float(input("Descuento (0 a 50): "))
while not 0 <= porcentaje <= 50:
    porcentaje = float(input("Descuento invalido: "))

descuento = precio * porcentaje / 100
total = precio - descuento
print(f"Descuento: {descuento:.2f}")
print(f"Total: {total:.2f}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
precio = 200; porcentaje = 25	Descuento = 200 × 25 / 100 = 50.
total	200 - 50 = 150.

Casos de prueba

200 y 0 → 200,00; 200 y 50 → 100,00; 200 y 25 → 150,00.

> **Error frecuente**
>
> Restar el porcentaje directamente, como si fuera un importe.

> **Desafío adicional**
>
> Incorporá un segundo descuento que se aplique sobre el precio ya rebajado.

MÓDULO 01 · EJERCICIO 005

05. Conversión de segundos

El problema

Ingresá un total de segundos entero no negativo. Convertí el valor a horas, minutos y segundos restantes.

Solución en Python

```python
total = int(input("Cantidad de segundos: "))
while total < 0:
    total = int(input("Ingrese un entero no negativo: "))

horas = total // 3600
resto = total % 3600
minutos = resto // 60
segundos = resto % 60

print(f"{horas:02d}:{minutos:02d}:{segundos:02d}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
total = 3661	Horas = 3661 // 3600 = 1; resto = 61.
resto = 61	Minutos = 1; segundos = 1. Salida 01:01:01.

Casos de prueba

0 → 00:00:00; 59 → 00:00:59; 3600 → 01:00:00; 3661 → 01:01:01.

Error frecuente

Usar división real / en lugar de división entera //.

Desafío adicional

Mostrá también la cantidad de días completos.

06. Calculadora de consumo de combustible

El problema

Leé una distancia recorrida en kilómetros y los litros de combustible utilizados. Ambos valores deben ser positivos y numéricos finitos. Calculá el consumo en litros cada 100 km y el rendimiento en km por litro.

Solución en Python

```python
from math import isfinite

distancia = float(input("Distancia en km: "))
litros = float(input("Litros consumidos: "))

while not (isfinite(distancia) and isfinite(litros) and distancia > 0 and litros > 0):
    print("Ambos valores deben ser positivos.")
    distancia = float(input("Distancia en km: "))
    litros = float(input("Litros consumidos: "))

consumo = litros / distancia * 100
rendimiento = distancia / litros
print(f"Consumo: {consumo:.2f} L/100 km")
print(f"Rendimiento: {rendimiento:.2f} km/L")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
distancia = 250; litros = 20	Consumo = 20 / 250 × 100 = 8.
rendimiento	250 / 20 = 12,50 km/L.

Casos de prueba

100 km y 5 L → 5,00 L/100 km; 250 km y 20 L → 8,00 L/100 km; distancia 0 → rechazar.

> **Error frecuente**
>
> Intercambiar el numerador y el denominador de las dos fórmulas.

> **Desafío adicional**
>
> Agregá el precio por litro y calculá el costo total del viaje.

MÓDULO 01 · EJERCICIO 007

07. Área y perímetro de un terreno

El problema

Solicitá el ancho y el largo de un terreno rectangular, expresados en metros. Aceptá únicamente valores numéricos finitos mayores que cero. Mostrá el área en metros cuadrados y el perímetro en metros.

Solución en Python

```python
from math import isfinite

ancho = float(input("Ancho (m): "))
largo = float(input("Largo (m): "))

while not (isfinite(ancho) and isfinite(largo) and ancho > 0 and largo > 0):
    print("Las dimensiones deben ser positivas.")
    ancho = float(input("Ancho (m): "))
    largo = float(input("Largo (m): "))

area = ancho * largo
perimetro = 2 * (ancho + largo)
print(f"Area: {area:.2f} m2")
print(f"Perimetro: {perimetro:.2f} m")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
ancho = 12; largo = 25	Área = 12 × 25 = 300 m².
perimetro	2 × (12 + 25) = 74 m.

Casos de prueba

5 y 5 → área 25; perímetro 20. 12 y 25 → área 300; perímetro 74. -2 y 8 → rechazar.

Error frecuente

Confundir el área con el perímetro o sumar los lados sin multiplicar por dos.

Desafío adicional

Calculá cuántos metros de alambre se necesitan para tres vueltas al terreno.

MÓDULO 01 · EJERCICIO 008

08. Presupuesto de materiales

El problema

Calculá cuántas cajas de revestimiento se necesitan para cubrir una superficie positiva. Cada caja cubre un área positiva. Suponé entradas numéricas finitas. Redondeá la cantidad de cajas hacia arriba y calculá el costo total con un precio por caja no negativo.

Solución en Python

```python
import math
from fractions import Fraction

superficie = Fraction(input("Superficie (m2): "))
cobertura = Fraction(input("Cobertura por caja: "))
precio = float(input("Precio por caja: "))

while superficie <= 0 or cobertura <= 0 or precio < 0:
    print("Datos invalidos. Reingrese.")
    superficie = Fraction(input("Superficie (m2): "))
    cobertura = Fraction(input("Cobertura por caja: "))
    precio = float(input("Precio por caja: "))

cajas = math.ceil(superficie / cobertura)
print(f"Cajas: {cajas}")
print(f"Total: {cajas * precio:.2f}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
superficie = 18; cobertura = 4	18 / 4 = 4,5; ceil redondea a 5 cajas.
precio = 1200	5 × 1200 = 6000.

Casos de prueba

18, 4, 1200 → 5 cajas y 6000,00. 20, 4, 100 → 5 cajas y 500,00.

Error frecuente

Usar int() para truncar el resultado y comprar material insuficiente.

Desafío adicional

Incluí un porcentaje de desperdicio antes de calcular las cajas.

09. Conversor de monedas con tasa fija

El problema

Ingresá un importe no negativo en una moneda de origen y una tasa positiva, expresada como unidades de destino por cada unidad de origen. Ambos datos son numéricos finitos. Mostrá el importe convertido con dos decimales. Usá valores hipotéticos, no cotizaciones reales.

Solución en Python

```python
importe = float(input("Importe de origen: "))
tasa = float(input("Tasa de conversion: "))

while importe < 0 or tasa <= 0:
    print("Importe o tasa invalidos.")
    importe = float(input("Importe de origen: "))
    tasa = float(input("Tasa de conversion: "))

convertido = importe * tasa
print(f"Importe convertido: {convertido:.2f}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
importe = 20; tasa = 4,5	Una unidad de origen equivale a 4,5 de destino.
convertido	20 × 4,5 = 90.

Casos de prueba

20 y 4,5 → 90,00. 0 y 2 → 0,00. Tasa 0 → rechazar.

Error frecuente

Dividir por la tasa cuando está expresada como unidades de destino por unidad de origen.

Desafío adicional

Agregá un menú para realizar también la conversión inversa.

10. Cálculo de IMC

El problema

Como ejercicio aritmético, ingresá una masa en kilogramos y una altura en metros, ambas numéricas finitas y positivas. Calculá el índice de masa corporal mediante masa / altura² y mostralo con dos decimales. No realices clasificaciones ni interpretaciones de salud.

Solución en Python

```python
masa = float(input("Masa (kg): "))
altura = float(input("Altura (m): "))

while masa <= 0 or altura <= 0:
    print("Los valores deben ser positivos.")
    masa = float(input("Masa (kg): "))
    altura = float(input("Altura (m): "))

imc = masa / altura ** 2
print(f"IMC calculado: {imc:.2f}")
```

Explicación y seguimiento manual

Paso / variable	Valor y explicación
masa = 72; altura = 1,80	El cuadrado de la altura es 3,24.
imc	72 / 3,24 = 22,22 aproximadamente.

Casos de prueba

72 y 1,80 → 22,22. 80 y 2 → 20,00. Altura 0 → rechazar.

Error frecuente

Utilizar centímetros en lugar de metros o dividir por la altura sin elevarla al cuadrado.

Desafío adicional

Permití ingresar la altura en centímetros y convertí el dato a metros.

CLUB PROGRAMADOR

¡Seguí aprendiendo!

Ya conociste los primeros 10 desafíos. La edición completa te acompaña desde los fundamentos hasta proyectos integradores.

- 142 páginas en PDF
- 100 ejercicios con soluciones explicadas
- 100 archivos Python para practicar
- Errores frecuentes y desafíos adicionales
- Apéndice de consulta rápida

Conseguí la edición completa

clubprogramador.com