

Taller Intensivo: Fundamentos de Python para el mejor curso

Información General

- **Duración:** 4 horas
 - **Nivel:** Fundamentos, bien fundamentados (Y)
 - **Requisitos:** Tener instalado Python 3.x y Visual Studio Code
-

Contenido del Taller

1. [Introducción y primer programa](#)
 2. [Variables y tipos de datos](#)
 3. [Operadores](#)
 4. [Entrada por teclado](#)
 5. [Condicionales](#)
 6. [Bucles](#)
 7. [Listas y diccionarios](#)
 8. [Funciones](#)
 9. [Uso de librerías estándar](#)
 10. [Mini-proyecto final](#)
-

1. Introducción y primer programa

Python es un lenguaje de programación conocido por su simplicidad y legibilidad. A diferencia del pseudocódigo, aquí escribiremos código real que la computadora puede ejecutar directamente. La gran ventaja es que la sintaxis de Python se parece mucho al pseudocódigo, lo que facilita la transición. En esta sección aprenderemos a escribir nuestro primer programa usando la función `print()` para mostrar mensajes en pantalla y a utilizar comentarios para documentar nuestro código.

Demostraciones

Ejemplo 1: Nuestro primer programa

```
# Este es mi primer programa en Python
print("¡Hola mundo!")
print("Bienvenidos al taller de Python")

# Podemos imprimir números directamente
print(2023)

# O hacer operaciones básicas
print(20 + 25)
```

Ejemplo 2: Comentarios en Python

```
# Esto es un comentario de una línea
print("Los comentarios son ignorados por Python") # También puedo comentar aquí
```

```
"""
Esto es un comentario
de múltiples líneas
Útil para explicaciones largas
"""

print("El programa continúa normalmente")
```

Ejercicios

Nivel Fácil

1. Escribe un programa que imprima tu nombre y edad en líneas separadas.
2. Crea un programa que imprima el nombre de tu canción favorita.

Nivel Medio

3. Escribe un programa que imprima un pequeño patrón como una carita feliz usando caracteres ASCII.
4. Crea un programa que imprima una cuenta regresiva del 5 al 1 (usando múltiples `print`).

Nivel Difícil

5. Escribe un programa con comentarios que imprima la letra de tu canción favorita con formato (espacios, saltos de línea, etc.).

Soluciones

```
# Ejercicio 1
print("Ana López")
print("19")

# Ejercicio 2
print("Perfect - Ed Sheeran")

# Ejercicio 3
print("  .-.")
print(" (o o)")
print("  \-/")
print("   =0=")

# Ejercicio 4
print("5")
print("4")
print("3")
print("2")
print("1")
print("¡Despegue!")

# Ejercicio 5
# Letra de "Despacito"
print("Sí, sabes que ya llevo un rato mirándote")
print("Tengo que bailar contigo hoy")
print("Vi que tu mirada ya estaba llamándome")
print("Muéstrame el camino que yo voy")
```

2. Variables y tipos de datos

Las variables son como cajas etiquetadas donde guardamos información que podemos usar después. En Python, no necesitamos declarar el tipo de variable antes de usarla. Trabajaremos con cuatro tipos básicos: enteros (int), decimales (float), texto (str) y booleanos (bool). Cada tipo de dato nos permite realizar diferentes operaciones y guardar distintos tipos de información.

Demostraciones

Ejemplo 1: Creando variables de diferentes tipos

```
# Variables numéricas
edad = 19 # Entero (int)
estatura = 1.75 # Decimal (float)

# Variables de texto
nombre = "Carlos" # Cadena de texto (str)
apellido = 'Gómez' # Las comillas simples también funcionan

# Variables booleanas
es_estudiante = True # Booleano (bool)
tiene_mascota = False

# Mostrando variables
print("Datos del usuario:")
print(nombre)
print(edad)
print(estatura)
print(es_estudiante)
```

Ejemplo 2: Modificando variables

```
# Creamos una variable
contador = 1
print("Contador inicial:", contador)

# Modificamos su valor
contador = 2
print("Contador modificado:", contador)

# Podemos usar el valor actual para calcular uno nuevo
contador = contador + 1
print("Contador incrementado:", contador)

# Asignación con operadores abreviados
contador += 5 # Equivalente a: contador = contador + 5
print("Contador después de sumar 5:", contador)
```

Ejemplo 3: Conversión entre tipos

```

# Convertir entre tipos de datos
numero_texto = "25"
numero = int(numero_texto) # Convertimos el texto a entero
print(numero + 5) # Ahora podemos hacer operaciones numéricas

edad = 19
edad_texto = str(edad) # Convertimos el número a texto
print("Tienes " + edad_texto + " años") # Podemos concatenar textos

precio = int(29.99) # Convertimos float a int (elimina decimales)
print(precio)

```

Ejercicios

Nivel Fácil

1. Crea variables para tu nombre, edad y estatura, luego imprímelas.
2. Crea una variable con tu año de nacimiento y otra con el año actual, luego calcula e imprime tu edad.

Nivel Medio

3. Crea una variable con el precio de un producto, aplícale un descuento del 15% y muestra el precio final.
4. Convierte tu edad a meses, semanas, días y horas, guardando cada resultado en una variable diferente.

Nivel Difícil

5. Crea variables para 3 notas de exámenes, calcula el promedio y muestra un mensaje que indique si aprobaste (promedio ≥ 7.0).

Soluciones

```

# Ejercicio 1
nombre = "Laura"
edad = 20
estatura = 1.65
print(nombre)
print(edad)
print(estatura)

# Ejercicio 2
anio_nacimiento = 2004
anio_actual = 2025
edad = anio_actual - anio_nacimiento
print(edad)

# Ejercicio 3
precio_original = 100
descuento = precio_original * 0.15
precio_final = precio_original - descuento
print(precio_final)

# Ejercicio 4

```

```

edad = 20
edad_meses = edad * 12
edad_semanas = edad * 52
edad_dias = edad * 365
edad_horas = edad_dias * 24
print(edad_meses)
print(edad_semanas)
print(edad_dias)
print(edad_horas)

# Ejercicio 5
nota1 = 8.5
nota2 = 6.0
nota3 = 7.5
promedio = (nota1 + nota2 + nota3) / 3
aprobado = promedio >= 7.0
print("Promedio:", promedio)
print("¿Aprobado?", aprobado)

```

3. Operadores

Los operadores son símbolos especiales que realizan operaciones con nuestras variables y valores. Python ofrece varios tipos de operadores que nos permiten realizar cálculos matemáticos, comparar valores y combinar condiciones lógicas. Dominar estos operadores es fundamental para crear programas que puedan tomar decisiones y procesar información.

Demostraciones

Ejemplo 1: Operadores aritméticos

```

# Operadores aritméticos básicos
a = 10
b = 3

suma = a + b # 13
resta = a - b # 7
multiplicacion = a * b # 30
division = a / b # 3.3333... (resultado decimal)
division_entera = a // b # 3 (división sin decimales)
modulo = a % b # 1 (resto de la división)
potencia = a ** b # 1000 (10 elevado a 3)

print("Suma:", suma)
print("Resta:", resta)
print("Multiplicación:", multiplicacion)
print("División:", division)
print("División entera:", division_entera)
print("Módulo:", modulo)
print("Potencia:", potencia)

```

Ejemplo 2: Operadores relacionales (comparación)

```

# Operadores de comparación
x = 7
y = 10

igual = x == y # False (¿x es igual a y?)
diferente = x != y # True (¿x es diferente de y?)
mayor = x > y # False (¿x es mayor que y?)
menor = x < y # True (¿x es menor que y?)
mayor_igual = x >= y # False (¿x es mayor o igual que y?)
menor_igual = x <= y # True (¿x es menor o igual que y?)

print("¿Iguales?", igual)
print("¿Diferentes?", diferente)
print("¿Mayor?", mayor)
print("¿Menor?", menor)
print("¿Mayor o igual?", mayor_igual)
print("¿Menor o igual?", menor_igual)

```

Ejemplo 3: Operadores lógicos

```

# Operadores lógicos
tiene_entrada = True
es_vip = False

# Operador AND (y)
puede_entrar_vip = tiene_entrada and es_vip # False
print("¿Puede entrar al área VIP?", puede_entrar_vip)

# Operador OR (o)
puede_entrar_evento = tiene_entrada or es_vip # True
print("¿Puede entrar al evento?", puede_entrar_evento)

# Operador NOT (no)
no_es_vip = not es_vip # True
print("¿No es VIP?", no_es_vip)

# Combinando operadores
edad = 16
con_adulto = True
puede_ver_pelicula = edad >= 18 or (edad >= 13 and con_adulto)
print("¿Puede ver la película?", puede_ver_pelicula)

```

Ejercicios

Nivel Fácil

1. Calcula el área y perímetro de un rectángulo con base 5 y altura 3.
2. Compara si 15 elevado a 2 es mayor que 200.

Nivel Medio

3. Crea un programa que determine si una persona con cierta edad puede votar (mayor o igual a 18) y tiene licencia de conducir.

4. Calcula el promedio de 4 exámenes y determina si el estudiante aprobó (promedio ≥ 7.0) y además asistió a más del 80% de las clases.

Nivel Difícil

5. Escribe un programa para una aplicación de streaming que determine si un usuario puede ver cierto contenido. Las condiciones son: ser mayor de 16 años, o mayor de 13 con permiso parental, y además tener una suscripción activa.

Soluciones

```
# Ejercicio 1
base = 5
altura = 3
area = base * altura
perimetro = 2 * (base + altura)
print("Área:", area)
print("Perímetro:", perimetro)

# Ejercicio 2
resultado = 15 ** 2 > 200
print("¿15 al cuadrado es mayor que 200?", resultado)

# Ejercicio 3
edad = 19
tiene_licencia = True
puede_conducir = edad >= 18 and tiene_licencia
print("¿Puede conducir?", puede_conducir)

# Ejercicio 4
nota1 = 8.0
nota2 = 7.5
nota3 = 6.0
nota4 = 9.0
promedio = (nota1 + nota2 + nota3 + nota4) / 4
asistencia = 85
aprobo = promedio >= 7.0 and asistencia > 80
print("Promedio:", promedio)
print("¿Aprobó?", aprobo)

# Ejercicio 5
edad_usuario = 14
permiso_padres = True
suscripcion_activa = True
puede_ver_contenido = (edad_usuario >= 16 or (edad_usuario >= 13 and permiso_padres))
and suscripcion_activa
print("¿Puede ver el contenido?", puede_ver_contenido)
```

4. Entrada por teclado

Para crear programas interactivos, necesitamos obtener información del usuario. Python nos permite hacer esto con la función `input()`, que detiene el programa y espera a que

el usuario escriba algo. Todo lo que se captura con `input()` se guarda como texto (string), así que si necesitamos números, deberemos convertirlos explícitamente.

Demostraciones

Ejemplo 1: Capturando texto

```
# Obtener texto del usuario
nombre = input("¿Cómo te llamas? ")
print("¡Hola,", nombre, "! Bienvenido/a al taller de Python.")

# También podemos capturar frases completas
frase = input("Escribe tu frase favorita: ")
print("Tu frase es:", frase)
print("Tu frase tiene", len(frase), "caracteres.")
```

Ejemplo 2: Capturando números

```
# Obtener un número y convertirlo
edad_texto = input("¿Cuántos años tienes? ")
edad = int(edad_texto) # Convertimos a entero
anio_nacimiento = 2025 - edad
print("Naciste aproximadamente en", anio_nacimiento)

# Podemos hacer la conversión directamente
estatura = float(input("¿Cuál es tu estatura en metros? "))
print("Tu estatura en centímetros es:", estatura * 100)
```

Ejemplo 3: Combinando entrada con operaciones

```
# Calculadora simple
print("CALCULADORA DE IMC (Índice de Masa Corporal)")
peso = float(input("Ingresa tu peso en kilogramos: "))
altura = float(input("Ingresa tu altura en metros: "))

# Cálculo del IMC
imc = peso / (altura ** 2)

print("Tu IMC es:", round(imc, 2))
print("Un IMC saludable está entre 18.5 y 24.9")
```

Ejercicios

Nivel Fácil

1. Crea un programa que pida el nombre de una película y el año de estreno, luego muestre cuántos años han pasado desde su estreno.
2. Escribe un programa que solicite un número y muestre su doble y su mitad.

Nivel Medio

3. Crea una calculadora que pida dos números y muestre la suma, resta, multiplicación y división de ambos.

4. Escribe un programa para una tienda que pida el precio de un producto y un porcentaje de descuento, y muestre el precio final.

Nivel Difícil

5. Desarrolla un conversor de tiempo que pida una cantidad de segundos y la convierta a horas, minutos y segundos (por ejemplo, 3665 segundos son 1 hora, 1 minuto y 5 segundos).

Soluciones

```
# Ejercicio 1
pelicula = input("Nombre de la película: ")
anio = int(input("Año de estreno: "))
anios_pasados = 2025 - anio
print(f"Han pasado {anios_pasados} años desde el estreno de {pelicula}.")

# Ejercicio 2
numero = float(input("Ingresa un número: "))
doble = numero * 2
mitad = numero / 2
print(f"El doble es {doble} y la mitad es {mitad}.")

# Ejercicio 3
num1 = float(input("Primer número: "))
num2 = float(input("Segundo número: "))
suma = num1 + num2
resta = num1 - num2
multiplicacion = num1 * num2
division = num1 / num2 if num2 != 0 else "No se puede dividir por cero"
print(f"Suma: {suma}")
print(f"Resta: {resta}")
print(f"Multiplicación: {multiplicacion}")
print(f"División: {division}")

# Ejercicio 4
precio = float(input("Precio del producto: $"))
descuento = float(input("Porcentaje de descuento: "))
monto_descuento = precio * (descuento / 100)
precio_final = precio - monto_descuento
print(f"Monto de descuento: ${monto_descuento}")
print(f"Precio final: ${precio_final}")

# Ejercicio 5
segundos_totales = int(input("Ingresa una cantidad de segundos: "))
horas = segundos_totales // 3600
minutos = (segundos_totales % 3600) // 60
segundos = segundos_totales % 60
print(f"{segundos_totales} segundos son {horas} horas, {minutos} minutos y {segundos} segundos.")
```

5. Condicionales

Las estructuras condicionales nos permiten tomar decisiones en nuestros programas. Con ellas, podemos ejecutar cierto código solo si se cumple una condición. En Python usamos `if`, `elif` y `else` para crear estas estructuras. El código dentro de cada bloque debe estar indentado (con espacios o tabulaciones) para indicar que pertenece a ese bloque.

Demostraciones

Ejemplo 1: Condicional simple (if)

```
# Estructura if básica
edad = int(input("¿Cuántos años tienes? "))

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

print("Este mensaje se muestra siempre")
```

Ejemplo 2: Condicional con alternativa (if-else)

```
# Estructura if-else
contrasenia = input("Ingresa tu contraseña: ")

if contrasenia == "python123":
    print("Contraseña correcta")
    print("Acceso concedido")
else:
    print("Contraseña incorrecta")
    print("Acceso denegado")

print("Fin del programa")
```

Ejemplo 3: Condicionales múltiples (if-elif-else)

```
# Estructura if-elif-else
calificacion = float(input("Ingresa tu calificación (0-10): "))

if calificacion >= 9:
    print("Excelente")
    print("¡Felicidades!")
elif calificacion >= 7:
    print("Bien")
    print("Aprobado")
elif calificacion >= 6:
    print("Suficiente")
    print("Apenas aprobado")
else:
    print("Insuficiente")
    print("Necesitas estudiar más")

print("Fin de la evaluación")
```

Ejercicios

Nivel Fácil

1. Escribe un programa que determine si un número ingresado por el usuario es par o impar.
2. Crea un programa que pida la edad del usuario y muestre un mensaje diferente para rangos: niño (0-12), adolescente (13-17), adulto (18-64) o adulto mayor (65+).

Nivel Medio

3. Desarrolla un programa para una tienda que calcule el descuento según el monto de compra: sin descuento hasta \$100, 10% hasta \$500, 20% hasta \$1000, 30% para montos mayores.
4. Crea un programa que pida tres números y determine cuál es el mayor.

Nivel Difícil

5. Escribe un programa que simule un cajero automático con un saldo inicial de \$1000. El usuario puede consultar saldo, depositar o retirar dinero. Si intenta retirar más del saldo disponible, debe mostrar un error.

Soluciones

```
# Ejercicio 1
numero = int(input("Ingresa un número: "))
if numero % 2 == 0:
    print("El número es par")
else:
    print("El número es impar")

# Ejercicio 2
edad = int(input("¿Cuál es tu edad? "))
if edad <= 12:
    print("Eres un niño")
elif edad <= 17:
    print("Eres un adolescente")
elif edad <= 64:
    print("Eres un adulto")
else:
    print("Eres un adulto mayor")

# Ejercicio 3
monto = float(input("Ingresa el monto de tu compra: $"))
if monto <= 100:
    descuento = 0
elif monto <= 500:
    descuento = monto * 0.1
elif monto <= 1000:
    descuento = monto * 0.2
else:
    descuento = monto * 0.3
precio_final = monto - descuento
print(f"Descuento: ${descuento}")
print(f"Total a pagar: ${precio_final}")
```

```

# Ejercicio 4
a = float(input("Primer número: "))
b = float(input("Segundo número: "))
c = float(input("Tercer número: "))
if a >= b and a >= c:
    mayor = a
elif b >= a and b >= c:
    mayor = b
else:
    mayor = c
print(f"El número mayor es {mayor}")

# Ejercicio 5
saldo = 1000
print("CAJERO AUTOMÁTICO")
print("1. Consultar saldo")
print("2. Depositar dinero")
print("3. Retirar dinero")
opcion = int(input("Elige una opción (1-3): "))

if opcion == 1:
    print(f"Tu saldo actual es: ${saldo}")
elif opcion == 2:
    deposito = float(input("¿Cuánto deseas depositar? $"))
    saldo += deposito
    print(f"Depósito exitoso. Nuevo saldo: ${saldo}")
elif opcion == 3:
    retiro = float(input("¿Cuánto deseas retirar? $"))
    if retiro <= saldo:
        saldo -= retiro
        print(f"Retiro exitoso. Nuevo saldo: ${saldo}")
    else:
        print("Error: Saldo insuficiente")
else:
    print("Opción no válida")

```

6. Bucles

Los bucles nos permiten repetir bloques de código varias veces. En Python tenemos principalmente dos tipos: `while` (mientras) que se ejecuta mientras una condición sea verdadera, y `for` (para) que se utiliza para recorrer elementos de una secuencia o repetir acciones un número determinado de veces. La función `range()` nos ayuda a generar secuencias de números para usar con `for`.

Demostraciones

Ejemplo 1: Bucle while

```

# Bucle while básico
contador = 1

```

```

while contador <= 5:
    print(f"Iteración {contador}")
    contador += 1 # Aumentamos el contador en cada iteración

print("El bucle ha terminado")

# While con condición de usuario
respuesta = ""

while respuesta != "salir":
    respuesta = input("Escribe 'salir' para terminar: ")
    print(f"Escribiste: {respuesta}")

print("Programa finalizado")

```

Ejemplo 2: Bucle for con range

```

# For con range
print("Contando del 1 al 5:")
for i in range(1, 6): # Del 1 al 5 (6 no se incluye)
    print(i)

print("Contando de 2 en 2:")
for i in range(0, 11, 2): # Del 0 al 10, de 2 en 2
    print(i)

print("Cuenta regresiva:")
for i in range(5, 0, -1): # Del 5 al 1, decrementando
    print(i)
print("¡Despegue!")

```

Ejemplo 3: For con colecciones

```

# For con strings
nombre = "Python"
print("Deletreando:")
for letra in nombre:
    print(letra)

# For con listas
amigos = ["Ana", "Juan", "Pedro", "Luis"]
print("Mis amigos son:")
for amigo in amigos:
    print(f"- {amigo}")

# For con índices
print("Amigos con posición:")
for indice in range(len(amigos)):
    print(f"{indice + 1}. {amigos[indice]}")

```

Ejercicios

Nivel Fácil

1. Escribe un programa que muestre la tabla de multiplicar del número que ingrese el usuario (del 1 al 10).
2. Crea un programa que solicite al usuario un número y luego muestre la suma de todos los números desde 1 hasta ese número.

Nivel Medio

3. Desarrolla un programa que calcule el factorial de un número ingresado por el usuario (factorial de $n = 1 \times 2 \times \dots \times n$).
4. Crea un programa que pida al usuario adivinar un número secreto entre 1 y 100, dándole pistas de "mayor" o "menor" hasta que lo adivine.

Nivel Difícil

5. Escribe un programa que genere la secuencia de Fibonacci hasta un límite ingresado por el usuario. La secuencia comienza con 0 y 1, y cada número posterior es la suma de los dos anteriores (0, 1, 1, 2, 3, 5, 8, 13, ...).

Soluciones

```
# Ejercicio 1
numero = int(input("Ingresa un número para ver su tabla de multiplicar: "))
print(f"Tabla del {numero}:")
for i in range(1, 11):
    resultado = numero * i
    print(f"{numero} x {i} = {resultado}")

# Ejercicio 2
limite = int(input("Ingresa un número: "))
suma = 0
for i in range(1, limite + 1):
    suma += i
print(f"La suma de los números del 1 al {limite} es: {suma}")

# Ejercicio 3
numero = int(input("Ingresa un número para calcular su factorial: "))
factorial = 1
for i in range(1, numero + 1):
    factorial *= i
print(f"El factorial de {numero} es: {factorial}")

# Ejercicio 4
import random
secreto = random.randint(1, 100)
intentos = 0
adivinado = False

print("Adivina un número entre 1 y 100")
while not adivinado:
    intento = int(input("Tu intento: "))
    intentos += 1

    if intento < secreto:
```

```

        print("El número secreto es MAYOR")
    elif intento > secreto:
        print("El número secreto es MENOR")
    else:
        print(f"¡Correcto! Lo adivinaste en {intentos} intentos")
        adivinado = True

# Ejercicio 5
limite = int(input("¿Hasta qué número quieres la secuencia de Fibonacci? "))
a, b = 0, 1
secuencia = [a, b]

while a + b <= limite:
    siguiente = a + b
    secuencia.append(siguiente)
    a, b = b, siguiente

print("Secuencia de Fibonacci:")
for numero in secuencia:
    print(numero, end=" ")

```

7. Listas y diccionarios

Las listas y diccionarios son estructuras que nos permiten almacenar múltiples valores en una sola variable. Las listas son colecciones ordenadas que pueden contener elementos de cualquier tipo, mientras que los diccionarios almacenan pares clave-valor, permitiéndonos acceder a los valores mediante sus claves asociadas.

Demostraciones

Ejemplo 1: Listas básicas

```

# Crear y acceder a elementos de una lista
amigos = ["Pedro", "Juan", "María", "Ana"]
print("Mi primer amigo es:", amigos[0]) # Pedro (los índices empiezan en 0)
print("Mi último amigo es:", amigos[-1]) # Ana (índice negativo cuenta desde el final)

# Modificar elementos
amigos[1] = "Carlos"
print("Lista actualizada:", amigos) # ["Pedro", "Carlos", "María", "Ana"]

# Añadir elementos
amigos.append("Luis") # Añade al final
print("Después de append:", amigos)

amigos.insert(1, "Diana") # Inserta en posición específica
print("Después de insert:", amigos)

# Eliminar elementos
eliminado = amigos.pop() # Elimina y devuelve el último elemento
print("Elemento eliminado:", eliminado)

```

```

print("Después de pop:", amigos)

amigos.remove("María") # Elimina por valor
print("Después de remove:", amigos)

# Longitud de una lista
print("Tengo", len(amigos), "amigos")

```

Ejemplo 2: Operaciones con listas

```

# Recorrer una lista
calificaciones = [9, 7, 8, 10, 6]
suma = 0

for calificacion in calificaciones:
    suma += calificacion

promedio = suma / len(calificaciones)
print(f"Calificaciones: {calificaciones}")
print(f"Promedio: {promedio}")

# Buscar en listas
if 10 in calificaciones:
    print("¡Tienes calificación perfecta!")

# Métodos útiles
print("Calificación más alta:", max(calificaciones))
print("Calificación más baja:", min(calificaciones))

calificaciones.sort() # Ordena la lista (modifica la original)
print("Calificaciones ordenadas:", calificaciones)

calificaciones.reverse() # Invierte la lista (modifica la original)
print("Calificaciones en orden descendente:", calificaciones)

# Sublistas (slicing)
primeras_tres = calificaciones[0:3] # Elementos del índice 0 al 2
print("Primeras tres calificaciones:", primeras_tres)

```

Ejemplo 3: Diccionarios

```

# Crear un diccionario
estudiante = {
    "nombre": "Ana García",
    "edad": 19,
    "carrera": "Ingeniería",
    "promedio": 8.7,
    "becado": True
}

# Acceder a valores
print("Nombre:", estudiante["nombre"])

```



```

print("Edad:", estudiante["edad"])

# Método alternativo para acceder (más seguro)
print("Carrera:", estudiante.get("carrera"))
print("Teléfono:", estudiante.get("telefono", "No registrado")) # Valor por defecto

# Modificar valores
estudiante["promedio"] = 9.1
print("Nuevo promedio:", estudiante["promedio"])

# Añadir nuevos pares clave-valor
estudiante["semestre"] = 3
print("Diccionario actualizado:", estudiante)

# Eliminar elementos
del estudiante["becado"]
print("Después de eliminar 'becado':", estudiante)

# Recorrer un diccionario
print("\nDatos del estudiante:")
for clave in estudiante:
    print(f"- {clave}: {estudiante[clave]}")

# Obtener claves y valores
print("Claves:", estudiante.keys())
print("Valores:", estudiante.values())

# Obtener pares clave-valor
for clave, valor in estudiante.items():
    print(f"{clave} = {valor}")

```

Ejercicios

Nivel Fácil

1. Crea una lista con los nombres de tus 5 películas favoritas y muestra la primera y la última.
2. Escribe un programa que cree un diccionario con información sobre una canción (título, artista, año, género) y muestre todos los datos.

Nivel Medio

3. Crea un programa que pida al usuario 5 números, los guarde en una lista, y luego muestre el mayor, el menor y el promedio.
4. Escribe un programa que lleve el registro de productos en un inventario usando un diccionario, donde las claves sean los nombres y los valores las cantidades. Permite al usuario añadir productos o actualizar cantidades.

Nivel Difícil

5. Desarrolla un programa que simule una agenda de contactos. Debe permitir agregar contactos (nombre y teléfono), buscar por nombre, mostrar todos los contactos y eliminar contactos.

Soluciones

```

# Ejercicio 1
peliculas = ["Interstellar", "Matrix", "El Padrino", "Pulp Fiction", "Inception"]
print("Primera película:", peliculas[0])
print("Última película:", peliculas[-1])

# Ejercicio 2
cancion = {
    "titulo": "Bohemian Rhapsody",
    "artista": "Queen",
    "año": 1975,
    "genero": "Rock"
}
print("Información de la canción:")
for clave, valor in cancion.items():
    print(f"{clave.capitalize()}: {valor}")

# Ejercicio 3
numeros = []
for i in range(5):
    numero = float(input(f"Ingresa el número {i+1}: "))
    numeros.append(numero)

mayor = max(numeros)
menor = min(numeros)
promedio = sum(numeros) / len(numeros)

print(f"Números ingresados: {numeros}")
print(f"Mayor: {mayor}")
print(f"Menor: {menor}")
print(f"Promedio: {promedio}")

# Ejercicio 4
inventario = {}

while True:
    print("\nINVENTARIO DE PRODUCTOS")
    print("1. Añadir/actualizar producto")
    print("2. Ver inventario")
    print("3. Salir")
    opcion = input("Elige una opción (1-3): ")

    if opcion == "1":
        nombre = input("Nombre del producto: ")
        cantidad = int(input("Cantidad: "))
        inventario[nombre] = cantidad
        print(f"Producto '{nombre}' actualizado con {cantidad} unidades")
    elif opcion == "2":
        if not inventario:
            print("El inventario está vacío")
        else:
            print("\nINVENTARIO ACTUAL:")
            for producto, cantidad in inventario.items():

```

```

        print(f"- {producto}: {cantidad} unidades")
    elif opcion == "3":
        print("Programa finalizado")
        break
    else:
        print("Opción no válida")

# Ejercicio 5
agenda = {}

while True:
    print("\nAGENDA DE CONTACTOS")
    print("1. Añadir contacto")
    print("2. Buscar contacto")
    print("3. Mostrar todos los contactos")
    print("4. Eliminar contacto")
    print("5. Salir")
    opcion = input("Elige una opción (1-5): ")

    if opcion == "1":
        nombre = input("Nombre del contacto: ")
        telefono = input("Número de teléfono: ")
        agenda[nombre] = telefono
        print(f"Contacto '{nombre}' agregado correctamente")
    elif opcion == "2":
        buscar = input("Ingresa el nombre a buscar: ")
        if buscar in agenda:
            print(f"Teléfono de {buscar}: {agenda[buscar]}")
        else:
            print(f"No se encontró el contacto '{buscar}'")
    elif opcion == "3":
        if not agenda:
            print("La agenda está vacía")
        else:
            print("\nLISTA DE CONTACTOS:")
            for nombre, telefono in agenda.items():
                print(f"- {nombre}: {telefono}")
    elif opcion == "4":
        eliminar = input("Ingresa el nombre del contacto a eliminar: ")
        if eliminar in agenda:
            del agenda[eliminar]
            print(f"Contacto '{eliminar}' eliminado correctamente")
        else:
            print(f"No se encontró el contacto '{eliminar}'")
    elif opcion == "5":
        print("¡Hasta pronto!")
        break
    else:
        print("Opción no válida")

```

8. Funciones

Las funciones son bloques de código reutilizables que realizan una tarea específica. Nos permiten organizar y modularizar nuestro código, evitando la repetición y facilitando el mantenimiento. En Python, creamos funciones con la palabra clave `def`, seguida del nombre de la función, paréntesis (que pueden contener parámetros) y dos puntos. Las funciones pueden devolver valores mediante la palabra clave `return`.

Demostraciones

Ejemplo 1: Funciones básicas

```
# Función sin parámetros
def saludar():
    print("¡Hola! Bienvenido al programa")
    print("Espero que te guste Python")

# Llamada a la función
saludar()

# Función con parámetros
def saludar_personalizado(nombre):
    print(f"¡Hola {nombre}! Bienvenido al programa")

# Llamadas con diferentes argumentos
saludar_personalizado("María")
saludar_personalizado("Carlos")

# Función con valor de retorno
def duplicar(numero):
    resultado = numero * 2
    return resultado

# Usar el valor retornado
x = duplicar(5)
print(f"El doble de 5 es: {x}")
print(f"El doble de 10 es: {duplicar(10)}")
```

Ejemplo 2: Funciones con múltiples parámetros y valores por defecto

```
# Función con múltiples parámetros
def calcular_promedio(a, b, c):
    suma = a + b + c
    promedio = suma / 3
    return promedio

resultado = calcular_promedio(8, 9, 7)
print(f"El promedio es: {resultado}")

# Función con parámetros opcionales (valores por defecto)
def saludar_formal(nombre, titulo="Sr.", despedida=False):
    mensaje = f"¡Hola {titulo} {nombre}!"
    if despedida:
        mensaje += " ¡Hasta pronto!"
```

```

    return mensaje

# Diferentes formas de llamar a la función
print(saludar_formal("García")) # Usa el valor por defecto para titulo
print(saludar_formal("López", "Dra. ")) # Cambia el titulo
print(saludar_formal("Martínez", "Ing.", True)) # Usa todos los parámetros

```

Ejemplo 3: Funciones como componentes de un programa

```

# Calculadora simple usando funciones
def sumar(a, b):
    return a + b

def restar(a, b):
    return a - b

def multiplicar(a, b):
    return a * b

def dividir(a, b):
    if b == 0:
        return "Error: División por cero"
    return a / b

# Programa principal
print("CALCULADORA")
print("1. Sumar")
print("2. Restar")
print("3. Multiplicar")
print("4. Dividir")

opcion = int(input("Elige una operación (1-4): "))
num1 = float(input("Primer número: "))
num2 = float(input("Segundo número: "))

if opcion == 1:
    resultado = sumar(num1, num2)
    operacion = "suma"
elif opcion == 2:
    resultado = restar(num1, num2)
    operacion = "resta"
elif opcion == 3:
    resultado = multiplicar(num1, num2)
    operacion = "multiplicación"
elif opcion == 4:
    resultado = dividir(num1, num2)
    operacion = "división"
else:
    resultado = "Opción no válida"
    operacion = "desconocida"

print(f"El resultado de la {operacion} es: {resultado}")

```

Ejercicios

Nivel Fácil

1. Crea una función que calcule y retorne el área de un círculo a partir de su radio.
2. Escribe una función que determine si un número es par o impar y devuelva True o False.

Nivel Medio

3. Desarrolla una función que convierta una temperatura en grados Celsius a Fahrenheit o viceversa, según un parámetro de modo ('C_a_F' o 'F_a_C').
4. Crea una función que reciba una lista de números y devuelva otra lista solo con los números pares.

Nivel Difícil

5. Escribe una función que determine si un número es primo o no (un número primo solo es divisible por 1 y por sí mismo).

Soluciones

```
# Ejercicio 1
def calcular_area_circulo(radio):
    pi = 3.14159
    area = pi * (radio ** 2)
    return area

radio = float(input("Ingresa el radio del círculo: "))
area = calcular_area_circulo(radio)
print(f"El área del círculo es: {area}")

# Ejercicio 2
def es_par(numero):
    return numero % 2 == 0

numero = int(input("Ingresa un número: "))
if es_par(numero):
    print(f"El número {numero} es par")
else:
    print(f"El número {numero} es impar")

# Ejercicio 3
def convertir_temperatura(temperatura, modo):
    if modo == "C_a_F":
        return (temperatura * 9/5) + 32
    elif modo == "F_a_C":
        return (temperatura - 32) * 5/9
    else:
        return "Modo no válido. Use 'C_a_F' o 'F_a_C'"

temp = float(input("Ingresa la temperatura: "))
modo = input("Ingresa el modo (C_a_F o F_a_C): ")
convertida = convertir_temperatura(temp, modo)
print(f"Temperatura convertida: {convertida}")
```

```

# Ejercicio 4
def obtener_pares(lista):
    pares = []
    for numero in lista:
        if numero % 2 == 0:
            pares.append(numero)
    return pares

# Probamos la función
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
pares = obtener_pares(numeros)
print(f"Lista original: {numeros}")
print(f"Solo pares: {pares}")

# Ejercicio 5
def es_primo(numero):
    # Los números menores o iguales a 1 no son primos
    if numero <= 1:
        return False
    # 2 es primo
    if numero == 2:
        return True
    # Los números pares mayores que 2 no son primos
    if numero % 2 == 0:
        return False

    # Comprobamos divisibilidad solo por números impares hasta la raíz cuadrada
    i = 3
    while i * i <= numero:
        if numero % i == 0:
            return False
        i += 2

    return True

num = int(input("Ingresa un número para verificar si es primo: "))
if es_primo(num):
    print(f"{num} es un número primo")
else:
    print(f"{num} no es un número primo")

```

9. Uso de librerías estándar

Python incluye una amplia variedad de librerías estándar que nos permiten realizar tareas comunes sin tener que escribir el código desde cero. Estas librerías abarcan desde matemáticas y estadísticas hasta manejo de archivos, procesamiento de datos y mucho más. En esta sección, exploraremos algunas de las librerías más útiles y cómo podemos aprovecharlas en nuestros programas.

Demostraciones

Ejemplo 1: Librería random

```
# Importar la librería
import random

# Generar un número aleatorio entre 1 y 100
numero = random.randint(1, 100)
print(f"Número aleatorio entre 1 y 100: {numero}")

# Elegir un elemento aleatorio de una lista
colores = ["rojo", "verde", "azul", "amarillo", "morado"]
color = random.choice(colores)
print(f"Color aleatorio: {color}")

# Barajar (mezclar) una lista
numeros = [1, 2, 3, 4, 5]
random.shuffle(numeros)
print(f"Lista barajada: {numeros}")

# Generar un número decimal aleatorio entre 0 y 1
decimal = random.random()
print(f"Número decimal aleatorio entre 0 y 1: {decimal}")

# Generar un número decimal aleatorio en un rango específico
precio = random.uniform(10.0, 50.0)
print(f"Precio aleatorio entre $10 y $50: ${precio:.2f}")
```

Ejemplo 2: Librería time

```
# Importar la librería
import time

# Obtener la hora actual
tiempo_actual = time.time() # Segundos desde el 1 de enero de 1970
print(f"Tiempo en segundos desde 'epoch': {tiempo_actual}")

# Convertir a una estructura de tiempo legible
tiempo_legible = time.localtime(tiempo_actual)
print(f"Año: {tiempo_legible.tm_year}")
print(f"Mes: {tiempo_legible.tm_mon}")
print(f"Día: {tiempo_legible.tm_mday}")
print(f"Hora: {tiempo_legible.tm_hour}:{tiempo_legible.tm_min}:{tiempo_legible.tm_sec}")

# Formato de fecha y hora
fecha_formateada = time.strftime("%d/%m/%Y %H:%M:%S", tiempo_legible)
print(f"Fecha y hora formateada: {fecha_formateada}")

# Pausar la ejecución del programa
print("El programa se detendrá por 2 segundos...")
time.sleep(2)
print("Continuamos después de la pausa")
```


Ejemplo 3: Librería math

```
# Importar la librería
import math

# Constantes matemáticas
print(f"Valor de Pi: {math.pi}")
print(f"Valor de e (Euler): {math.e}")

# Funciones trigonométricas (en radianes)
angulo = math.pi / 4 # 45 grados
print(f"Seno de 45°: {math.sin(angulo)}")
print(f"Coseno de 45°: {math.cos(angulo)}")
print(f"Tangente de 45°: {math.tan(angulo)}")

# Redondeo
print(f"Redondeo hacia arriba de 4.3: {math.ceil(4.3)}")
print(f"Redondeo hacia abajo de 4.7: {math.floor(4.7)}")
print(f"Redondeo al más cercano de 4.5: {round(4.5)}")

# Otras funciones útiles
print(f"Raíz cuadrada de 16: {math.sqrt(16)}")
print(f"5 elevado a la 3: {math.pow(5, 3)}")
print(f"Logaritmo natural de 10: {math.log(10)}")
print(f"Logaritmo base 10 de 100: {math.log10(100)}")
print(f"Valor absoluto de -7.5: {math.fabs(-7.5)}")
```

Ejercicios

Nivel Fácil

1. Usando la librería `random`, crea un programa que simule el lanzamiento de un dado (números del 1 al 6) cinco veces.
2. Con la librería `time`, haz un programa que muestre la fecha y hora actual en el formato "Hoy es [día] de [mes] de [año], y son las [hora]:[minutos]".

Nivel Medio

3. Crea un programa que genere una contraseña aleatoria de longitud variable (dada por el usuario) usando letras, números y símbolos.
4. Utilizando la librería `math`, escribe un programa que calcule el área y perímetro de un círculo, pidiendo al usuario el radio.

Nivel Difícil

5. Desarrolla un juego de "Piedra, Papel o Tijera" donde el usuario juegue contra la computadora. El juego debe continuar hasta que uno de los dos gane 3 rondas.

Soluciones

```
# Ejercicio 1
import random

print("Simulando 5 lanzamientos de dado:")
for i in range(5):
```

```

    dado = random.randint(1, 6)
    print(f"Lanzamiento {i+1}: {dado}")

# Ejercicio 2
import time

ahora = time.localtime()
dia = ahora.tm_mday
mes = ["enero", "febrero", "marzo", "abril", "mayo", "junio",
       "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"]
[ahora.tm_mon - 1]
año = ahora.tm_year
hora = ahora.tm_hour
minutos = ahora.tm_min

print(f"Hoy es {dia} de {mes} de {año}, y son las {hora}:{minutos:02d}")

# Ejercicio 3
import random
import string

def generar_contraseña(longitud):
    # Definimos los caracteres posibles
    letras = string.ascii_letters # a-z, A-Z
    numeros = string.digits # 0-9
    simbolos = string.punctuation # !@#$%^&*()_+, etc.

    # Combinamos todos los caracteres
    todos_caracteres = letras + numeros + simbolos

    # Generamos la contraseña aleatoria
    contraseña = ''.join(random.choice(todos_caracteres) for i in range(longitud))
    return contraseña

longitud = int(input("¿De qué longitud quieres la contraseña? "))
password = generar_contraseña(longitud)
print(f"Tu contraseña generada es: {password}")

# Ejercicio 4
import math

def calcular_circulo(radio):
    area = math.pi * radio ** 2
    perimetro = 2 * math.pi * radio
    return area, perimetro

radio = float(input("Ingresa el radio del círculo: "))
area, perimetro = calcular_circulo(radio)

print(f"Para un círculo de radio {radio}:")
print(f"Área: {area:.2f} unidades cuadradas")
print(f"Perímetro: {perimetro:.2f} unidades")

```

```

# Ejercicio 5
import random
import time

def obtener_eleccion_computadora():
    return random.choice(["piedra", "papel", "tijera"])

def determinar_ganador(usuario, computadora):
    if usuario == computadora:
        return "Empate"

    if (usuario == "piedra" and computadora == "tijera") or \
        (usuario == "papel" and computadora == "piedra") or \
        (usuario == "tijera" and computadora == "papel"):
        return "Usuario"
    else:
        return "Computadora"

print("JUEGO DE PIEDRA, PAPEL O TIJERA")
print("El primero en ganar 3 rondas, gana el juego.")

puntos_usuario = 0
puntos_computadora = 0
ronda = 0

while puntos_usuario < 3 and puntos_computadora < 3:
    ronda += 1
    print(f"\nRonda {ronda} - Puntuación: Usuario {puntos_usuario}, Computadora {puntos_computadora}")

    # Elección del usuario
    eleccion_usuario = input("Elige piedra, papel o tijera: ").lower()
    while eleccion_usuario not in ["piedra", "papel", "tijera"]:
        eleccion_usuario = input("Entrada no válida. Elige piedra, papel o tijera: ").lower()

    # Elección de la computadora
    eleccion_computadora = obtener_eleccion_computadora()

    # Agregamos dramatismo
    print("La computadora está eligiendo", end="")
    for _ in range(3):
        print(".", end="", flush=True)
        time.sleep(0.5)
    print()

    print(f"Tu elección: {eleccion_usuario}")
    print(f"Elección de la computadora: {eleccion_computadora}")

    # Determinar ganador
    resultado = determinar_ganador(eleccion_usuario, eleccion_computadora)

```

```

if resultado == "Usuario":
    print("¡Ganaste esta ronda!")
    puntos_usuario += 1
elif resultado == "Computadora":
    print("La computadora gana esta ronda.")
    puntos_computadora += 1
else:
    print("Esta ronda es un empate.")

# Fin del juego
print("\n" + "=" * 30)
print(f"Puntuación final: Usuario {puntos_usuario}, Computadora {puntos_computadora}")

if puntos_usuario > puntos_computadora:
    print("¡Felicidades! Has ganado el juego.")
else:
    print("La computadora ha ganado el juego. ¡Suerte la próxima vez!")

```

10. Mini-proyecto final

Después de haber aprendido los fundamentos de Python, vamos a poner en práctica todo lo aprendido con un pequeño proyecto que integra varios conceptos. Este proyecto es opcional pero altamente recomendado para consolidar tus conocimientos.

Juego de Trivia

Crearemos un juego de preguntas y respuestas que:

1. Tenga una variedad de preguntas sobre diferentes temas
2. Lleve un registro de puntos
3. Permita al usuario elegir categorías
4. Muestre retroalimentación inmediata

```

import random
import time

def presentar_juego():
    """Muestra la presentación del juego"""
    print("=" * 50)
    print("BIENVENIDO A LA SUPER TRIVIA PYTHON".center(50))
    print("=" * 50)
    print("\nPon a prueba tus conocimientos generales en este juego de trivia.")
    print("Cada respuesta correcta vale 10 puntos.")
    print("\n¡Vamos a comenzar!")
    time.sleep(1)

def obtener_nombre():
    """Solicita y retorna el nombre del jugador"""
    while True:
        nombre = input("\n¿Cuál es tu nombre? ")
        if nombre.strip(): # Verifica que no esté vacío

```

```

        return nombre
    print("Por favor, ingresa un nombre válido.")

def mostrar_categorias():
    """Muestra las categorías disponibles y retorna la elección del usuario"""
    print("\nCATEGORÍAS DISPONIBLES:")
    print("1. Ciencia")
    print("2. Historia")
    print("3. Deportes")
    print("4. Entretenimiento")
    print("5. Todas las categorías")

    while True:
        try:
            eleccion = int(input("\nElige una categoría (1-5): "))
            if 1 <= eleccion <= 5:
                return eleccion
            else:
                print("Por favor, elige un número entre 1 y 5.")
        except ValueError:
            print("Por favor, ingresa un número válido.")

def obtener_preguntas(categoria):
    """Retorna las preguntas según la categoría elegida"""
    # Base de datos de preguntas por categoría
    preguntas = {
        "Ciencia": [
            {
                "pregunta": "¿Cuál es el planeta más grande del sistema solar?",
                "opciones": ["A. Tierra", "B. Júpiter", "C. Saturno", "D. Neptuno"],
                "respuesta": "B"
            },
            {
                "pregunta": "¿Cuál es el símbolo químico del oro?",
                "opciones": ["A. Au", "B. Ag", "C. Fe", "D. Or"],
                "respuesta": "A"
            },
            {
                "pregunta": "¿Qué gas respiran las plantas en la fotosíntesis?",
                "opciones": ["A. Oxígeno", "B. Nitrógeno", "C. Dióxido de carbono",
"D. Helio"],
                "respuesta": "C"
            }
        ],
        "Historia": [
            {
                "pregunta": "¿En qué año llegó Cristóbal Colón a América?",
                "opciones": ["A. 1492", "B. 1592", "C. 1392", "D. 1500"],
                "respuesta": "A"
            },
            {
                "pregunta": "¿Quién fue el primer presidente de Estados Unidos?",

```

```

        "opciones": ["A. Thomas Jefferson", "B. Abraham Lincoln", "C. George
Washington", "D. John Adams"],
        "respuesta": "C"
    },
    {
        "pregunta": "¿Qué imperio construyó Machu Picchu?",
        "opciones": ["A. Maya", "B. Azteca", "C. Inca", "D. Olmeca"],
        "respuesta": "C"
    }
],
"Deportes": [
    {
        "pregunta": "¿Cuántos jugadores tiene un equipo de fútbol en el
campo?",
        "opciones": ["A. 9", "B. 10", "C. 11", "D. 12"],
        "respuesta": "C"
    },
    {
        "pregunta": "¿Qué deporte se juega en Wimbledon?",
        "opciones": ["A. Golf", "B. Tenis", "C. Cricket", "D. Rugby"],
        "respuesta": "B"
    },
    {
        "pregunta": "¿Cada cuántos años se celebran los Juegos Olímpicos?",
        "opciones": ["A. 2 años", "B. 3 años", "C. 4 años", "D. 5 años"],
        "respuesta": "C"
    }
],
"Entretenimiento": [
    {
        "pregunta": "¿Quién interpretó a Iron Man en el Universo
Cinematográfico de Marvel?",
        "opciones": ["A. Chris Evans", "B. Robert Downey Jr.", "C. Chris
Hemsworth", "D. Mark Ruffalo"],
        "respuesta": "B"
    },
    {
        "pregunta": "¿Cuál es la saga de libros más vendida de la historia?",
        "opciones": ["A. El Señor de los Anillos", "B. Harry Potter", "C.
Crepúsculo", "D. Percy Jackson"],
        "respuesta": "B"
    },
    {
        "pregunta": "¿Qué banda cantó 'Bohemian Rhapsody'?",
        "opciones": ["A. The Beatles", "B. Led Zeppelin", "C. Queen", "D. Pink
Floyd"],
        "respuesta": "C"
    }
]
}

```

Seleccionar preguntas según la categoría elegida

```

if categoria == 1:
    return preguntas["Ciencia"]
elif categoria == 2:
    return preguntas["Historia"]
elif categoria == 3:
    return preguntas["Deportes"]
elif categoria == 4:
    return preguntas["Entretenimiento"]
else: # Todas las categorías
    todas = []
    for cat in preguntas.values():
        todas.extend(cat)
    random.shuffle(todas)
    return todas[:5] # Limitar a 5 preguntas aleatorias

def hacer_pregunta(pregunta, num_pregunta):
    """Muestra una pregunta, sus opciones y evalúa la respuesta"""
    print(f"\nPregunta {num_pregunta}:")
    print(pregunta["pregunta"])

    for opcion in pregunta["opciones"]:
        print(opcion)

    while True:
        respuesta = input("\nTu respuesta (A, B, C o D): ").upper()
        if respuesta in ["A", "B", "C", "D"]:
            break
        print("Por favor, ingresa una opción válida (A, B, C o D).")

    es_correcta = respuesta == pregunta["respuesta"]

    if es_correcta:
        print("\n¡Correcto! +10 puntos")
        return 10
    else:
        print(f"\nIncorrecto. La respuesta correcta era: {pregunta['respuesta']}")
        return 0

def mostrar_resultado(nombre, puntos, total_preguntas):
    """Muestra el resultado final del juego"""
    print("\n" + "=" * 50)
    print(f"RESULTADO FINAL PARA {nombre.upper()}.center(50))
    print("=" * 50)
    print(f"\nPuntuación total: {puntos} de {total_preguntas * 10} puntos posibles")

    # Calcular porcentaje de aciertos
    porcentaje = (puntos / (total_preguntas * 10)) * 100
    print(f"Porcentaje de aciertos: {porcentaje:.1f}%")

    # Mensajes según el rendimiento
    if porcentaje >= 80:
        print("\n¡Excelente! Eres un maestro del conocimiento general.")

```

```

elif porcentaje >= 60:
    print("\n¡Muy bien! Tienes buenos conocimientos generales.")
elif porcentaje >= 40:
    print("\nBien hecho. Sigue aprendiendo para mejorar tu puntuación.")
else:
    print("\nSigue practicando, ¡la próxima vez lo harás mejor!")

def jugar_de_nuevo():
    """Pregunta al usuario si desea jugar otra partida"""
    respuesta = input("\n¿Quieres jugar otra partida? (s/n): ").lower()
    return respuesta == 's'

def main():
    """Función principal que controla el flujo del juego"""
    presentar_juego()
    nombre = obtener_nombre()

    jugar = True
    while jugar:
        # Seleccionar categoría
        categoria = mostrar_categorias()
        preguntas = obtener_preguntas(categoria)

        # Inicializar puntuación
        puntos = 0

        # Jugar ronda de preguntas
        for i, pregunta in enumerate(preguntas, 1):
            puntos += hacer_pregunta(pregunta, i)
            time.sleep(1) # Pequeña pausa entre preguntas

        # Mostrar resultados
        mostrar_resultado(nombre, puntos, len(preguntas))

        # Preguntar si quiere jugar otra vez
        jugar = jugar_de_nuevo()

    print(f"\n¡Gracias por jugar, {nombre}! Hasta la próxima.")

if __name__ == "__main__":
    main()

```