**Programación 3**

****

|  |  |
| --- | --- |
| **Tarea #: 1** |  |

**Autora: Mónica Rea**

**Carrera: Tecnología Superior en Desarrollo de Software**

**Fecha: 21/11/2024**

**Introducción**

La programación es una actividad que implica escribir instrucciones en un lenguaje comprensible por las computadoras, para que estas realicen tareas específicas. A través de la programación, los desarrolladores crean programas y aplicaciones que permiten automatizar procesos, realizar cálculos complejos o interactuar con sistemas de manera eficiente. Los lenguajes de programación, como Python y Java, ofrecen herramientas y estructuras que permiten estructurar el código de forma lógica, facilitando su comprensión y mantenimiento (Zhao et al., 2020). La programación también fomenta habilidades como la resolución de problemas y el pensamiento crítico, ya que implica analizar problemas y descomponerlos en pasos más pequeños y manejables (McConnell, 2004).

En Python y Java, existen muchos ejercicios sencillos que los principiantes pueden practicar para familiarizarse con los conceptos básicos de la programación. Un ejemplo común es escribir un programa que calcule la suma de dos números. En Python, esto puede lograrse fácilmente con una simple línea de código: suma = numero1 + numero2. Otro ejercicio básico es crear una calculadora que realice operaciones aritméticas simples, lo cual ayuda a entender cómo trabajar con variables y operadores (Van der Veen et al., 2017). Asimismo, en Java, un programa sencillo que imprime "Hola, Mundo" al usuario es uno de los primeros pasos que los novatos dan al comenzar a programar, y sirve para entender la estructura básica de un programa en este lenguaje.

Otro ejercicio accesible para los principiantes es crear un programa que pida al usuario su edad y determine si es mayor de edad o no. Este tipo de ejercicio ayuda a comprender el uso de condicionales (if y else), que son fundamentales para tomar decisiones dentro de un programa. Además, realizar este tipo de prácticas permite a los aprendices ver resultados inmediatos, lo cual mantiene el interés y facilita el aprendizaje de conceptos como el manejo de entradas y salidas de datos (Zhao et al., 2020). Al continuar practicando con ejercicios más complejos, los programadores novatos pueden desarrollar habilidades más avanzadas, como la creación de interfaces gráficas o la manipulación de bases de datos.

Resolver

1. **Mayor o menor:**
   * Escribe un programa que solicite un número y determine si es mayor o menor que 10.
   * Ejemplo: Entrada: 5 → Salida: *"Es menor que 10"*.
2. **Número positivo o negativo:**
   * Solicita al usuario un número y determina si es positivo, negativo o cero.
   * Ejemplo: Entrada: -3 → Salida: *"Es un número negativo"*.
3. **Par o impar:**
   * Solicita un número al usuario y determina si es par o impar.
   * Ejemplo: Entrada: 4 → Salida: *"Es par"*.
4. **Aprobado o reprobado:**
   * Solicita la calificación de un estudiante y determina si está aprobado (mayor o igual a 7) o reprobado.
   * Ejemplo: Entrada: 6.5 → Salida: *"Reprobado"*.

**Ejercicios intermedios**

1. **Descuento en una tienda:**
   * Una tienda ofrece un 20% de descuento si el cliente gasta más de $100. Escribe un programa que calcule el monto final.
   * Ejemplo: Entrada: $120 → Salida: *"Monto final: $96"*.
2. **Edad para votar:**
   * Solicita la edad del usuario y determina si es elegible para votar (mayor o igual a 18 años).
   * Ejemplo: Entrada: 17 → Salida: *"No puedes votar"*.
3. **Mayor de tres números:**
   * Solicita tres números y determina cuál es el mayor.
   * Ejemplo: Entrada: 4, 9, 2 → Salida: *"El número mayor es 9"*.
4. **Clasificación de edades:**
   * Solicita una edad y clasifica al usuario como niño (0-12), adolescente (13-17) o adulto (18+).
   * Ejemplo: Entrada: 15 → Salida: *"Eres adolescente"*.
5. **Calculadora básica:**
   * Solicita dos números y una operación (+, -, \*, /) y realiza el cálculo correspondiente.
   * Ejemplo: Entrada: 3, 2, '+' → Salida: *"Resultado: 5"*.
6. **Determinar un año bisiesto:**
   * Solicita un año y determina si es bisiesto (divisible entre 4 pero no entre 100, excepto si es divisible entre 400).
   * Ejemplo: Entrada: 2024 → Salida: *"Es bisiesto"*.
7. **Validar contraseñas:**
   * Escribe un programa que solicite una contraseña y valide si es correcta (ejemplo: contraseña fija es 12345).
   * Ejemplo: Entrada: 12345 → Salida: *"Acceso concedido"*.
8. **Juego de números:**
   * Genera un número aleatorio entre 1 y 10 y solicita al usuario que adivine el número. Usa if para verificar si acertó o no.
   * Ejemplo: Entrada: 5 → Salida: *"¡Felicidades, acertaste!"* o *"Intenta de nuevo."*.
9. **Calcular el signo zodiacal:**
   * Solicita el día y mes de nacimiento y determina el signo zodiacal del usuario.
   * Ejemplo: Entrada: 22, marzo → Salida: *"Tu signo es Aries"*.
10. **Sistema de calificaciones:**
    * Solicita una calificación numérica y devuelve la letra correspondiente:
      + 90-100: A.
      + 80-89: B.
      + 70-79: C.
      + 60-69: D.
      + Menor a 60: F.
    * Ejemplo: Entrada: 85 → Salida: *"Tu calificación es B"*.
11. **Control de acceso:**
    * Solicita un nombre de usuario y contraseña, y valida si ambos son correctos. Permite tres intentos antes de bloquear el acceso.
    * Ejemplo: Entrada: Usuario: admin, Contraseña: 1234 → Salida: *"Bienvenido, admin."*.

Bucles

1. **Imprimir números del 1 al 10:**
   * Usa un bucle para mostrar los números del 1 al 10 en la consola.
   * **Salida esperada:** 1, 2, 3, ..., 10.
2. **Suma de los primeros 10 números:**
   * Calcula y muestra la suma de los números del 1 al 10.
   * **Salida esperada:** 55.
3. **Tabla de multiplicar:**
   * Solicita un número al usuario y muestra su tabla de multiplicar del 1 al 10.
   * **Ejemplo:** Entrada: 5 → Salida: 5 x 1 = 5, ..., 5 x 10 = 50.
4. **Contar números pares:**
   * Imprime todos los números pares entre 1 y 20.
   * **Salida esperada:** 2, 4, 6, ..., 20.
5. **Contador regresivo:**
   * Imprime los números del 10 al 1 en orden descendente.
   * **Salida esperada:** 10, 9, 8, ..., 1.

**Ejercicios intermedios**

1. **Factorial de un número:**
   * Calcula el factorial de un número ingresado por el usuario (n!).
   * **Ejemplo:** Entrada: 5 → Salida: 120.
2. **Números primos:**
   * Encuentra e imprime todos los números primos entre 1 y 50.
   * **Salida esperada:** 2, 3, 5, 7, ..., 47.
3. **Sumar dígitos:**
   * Solicita un número entero y calcula la suma de sus dígitos.
   * **Ejemplo:** Entrada: 123 → Salida: 6 (1 + 2 + 3).
4. **Invertir un número:**
   * Solicita un número entero y muestra su versión invertida.
   * **Ejemplo:** Entrada: 1234 → Salida: 4321.
5. **Promedio de calificaciones:**
   * Solicita calificaciones al usuario (hasta que ingrese -1) y calcula el promedio.
   * **Ejemplo:** Entradas: 5, 7, 8, -1 → Salida: Promedio: 6.67.

Funciones

1. **Saludo personalizado:**
   * Crea una función que reciba un nombre como parámetro y retorne un saludo.
   * **Ejemplo:** Entrada: María → Salida: *"Hola, María!"*.
2. **Suma de dos números:**
   * Escribe una función que reciba dos números como parámetros y retorne su suma.
   * **Ejemplo:** Entrada: 3, 7 → Salida: 10.
3. **Número par o impar:**
   * Crea una función que reciba un número y retorne True si es par y False si es impar.
   * **Ejemplo:** Entrada: 4 → Salida: True.
4. **Calcular el cuadrado:**
   * Escribe una función que reciba un número y retorne su cuadrado.
   * **Ejemplo:** Entrada: 5 → Salida: 25.
5. **Calcular el área de un círculo:**
   * Crea una función que reciba el radio de un círculo y retorne su área.
   * Fórmula: Área = π \* radio^2.
   * **Ejemplo:** Entrada: 3 → Salida: 28.27 (aproximado).

**Desarrollo**

**Ejercicio 1.** Mayor o menor:

Escribe un programa que solicite un número y determine si es mayor o menor que 10. Ejemplo: Entrada: 5 → Salida: "Es menor que 10".

En Python

# Solicitar al usuario que ingrese un número

numero = float(input("Ingresa un número: "))

# Verificar si el número es mayor o menor que 10

if numero > 10:

print("Es mayor que 10")

elif numero < 10:

print("Es menor que 10")

else:

print("Es igual a 10")

En java

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

// Crear un objeto Scanner para leer la entrada del usuario

Scanner scanner = new Scanner(System.in);

// Solicitar al usuario que ingrese un número

System.out.print("Ingresa un número: ");

double numero = scanner.nextDouble();

// Verificar si el número es mayor, menor o igual a 10

if (numero > 10) {

System.out.println("Es mayor que 10");

} else if (numero < 10) {

System.out.println("Es menor que 10");

} else {

System.out.println("Es igual a 10");

}

// Cerrar el scanner

scanner.close();

}

}

Ejercicio 2. Número positivo o negativo:

Solicita al usuario un número y determina si es positivo, negativo o cero. Ejemplo: Entrada: -3 → Salida: "Es un número negativo"

En Python

# Solicitar al usuario que ingrese un número

numero = float(input("Ingresa un número: "))

# Verificar si el número es positivo, negativo o cero

if numero > 0:

print("Es un número positivo")

elif numero < 0:

print("Es un número negativo")

else:

print("Es cero")

En java

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

3. Par o impar:

Solicita un número al usuario y determina si es par o impar. Ejemplo: Entrada: 4 → Salida: "Es par"

# Programa para determinar si un número es par o impar

try:

# Solicitar al usuario un número entero

numero = int(input("Introduce un número entero: "))

# Verificar si el número es par o impar

if numero % 2 == 0:

print("Es par")

else:

print("Es impar")

except ValueError:

print("Por favor, introduce un número entero válido.")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

4. Aprobado o reprobado:

Solicita la calificación de un estudiante y determina si está aprobado (mayor o igual a 7) o reprobado. Ejemplo: Entrada: 6.5 → Salida: "Reprobado".

# Programa para determinar si un estudiante está aprobado o reprobado

try:

# Solicitar al usuario la calificación del estudiante

calificacion = float(input("Introduce la calificación del estudiante: "))

# Verificar si está aprobado o reprobado

if calificacion >= 7:

print("Aprobado")

else:

print("Reprobado")

except ValueError:

print("Por favor, introduce una calificación válida.")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ejercicios intermedios**

5. Descuento en una tienda:

Una tienda ofrece un 20% de descuento si el cliente gasta más de $100. Escribe un programa que calcule el monto final. Ejemplo: Entrada: $120 → Salida: "Monto final: $96"

# Programa para calcular el monto final con descuento

try:

# Solicitar al usuario el monto gastado

monto = float(input("Introduce el monto gastado: $"))

# Verificar si aplica el descuento

if monto > 100:

descuento = monto \* 0.20

monto\_final = monto - descuento

else:

descuento = 0

monto\_final = monto

# Mostrar el resultado

print(f"Monto final: ${monto\_final:.2f}")

except ValueError:

print("Por favor, introduce un monto válido.")

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

6. Edad para votar:

Solicita la edad del usuario y determina si es elegible para votar (mayor o igual a 18 años). Ejemplo: Entrada: 17 → Salida: "No puedes votar"

# Programa para verificar si un usuario es elegible para votar

try:

# Solicitar la edad del usuario

edad = int(input("Introduce tu edad: "))

# Verificar si puede votar

if edad >= 18:

print("Puedes votar")

else:

print("No puedes votar")

except ValueError:

print("Por favor, introduce una edad válida.")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

7. Mayor de tres números:

Solicita tres números y determina cuál es el mayor. Ejemplo: Entrada: 4, 9, 2 → Salida: "El número mayor es 9"

# Programa para encontrar el mayor de tres números

try:

# Solicitar al usuario tres números

numero1 = float(input("Introduce el primer número: "))

numero2 = float(input("Introduce el segundo número: "))

numero3 = float(input("Introduce el tercer número: "))

# Determinar el mayor de los tres números

mayor = max(numero1, numero2, numero3)

# Mostrar el resultado

print(f"El número mayor es {mayor}")

except ValueError:

print("Por favor, introduce valores numéricos válidos.")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

8. Clasificación de edades:

Solicita una edad y clasifica al usuario como niño (0-12), adolescente (13-17) o adulto (18+). Ejemplo: Entrada: 15 → Salida: "Eres adolescente"

# Programa para clasificar edades

try:

# Solicitar la edad del usuario

edad = int(input("Introduce tu edad: "))

# Clasificar la edad

if 0 <= edad <= 12:

print("Eres niño")

elif 13 <= edad <= 17:

print("Eres adolescente")

elif edad >= 18:

print("Eres adulto")

else:

print("La edad no puede ser negativa. Por favor, introduce una edad válida.")

except ValueError:

print("Por favor, introduce una edad válida.")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9. Calculadora básica:

Solicita dos números y una operación (+, -, \*, /) y realiza el cálculo correspondiente. Ejemplo: Entrada: 3, 2, '+' → Salida: "Resultado: 5"

def calculadora():

print("Calculadora básica en Python")

# Solicitar números al usuario

try:

numero1 = float(input("Introduce el primer número: "))

numero2 = float(input("Introduce el segundo número: "))

except ValueError:

print("Por favor, introduce valores numéricos.")

return

# Solicitar la operación

operacion = input("Introduce la operación (+, -, \*, /): ")

# Realizar la operación

if operacion == '+':

resultado = numero1 + numero2

elif operacion == '-':

resultado = numero1 - numero2

elif operacion == '\*':

resultado = numero1 \* numero2

elif operacion == '/':

if numero2 != 0:

resultado = numero1 / numero2

else:

print("Error: No se puede dividir entre cero.")

return

else:

print("Operación no válida.")

return

# Mostrar el resultado

print(f"Resultado: {resultado}")

# Ejecutar la calculadora

calculadora()

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

10. Determinar un año bisiesto:

Solicita un año y determina si es bisiesto (divisible entre 4 pero no entre 100, excepto si es divisible entre 400). Ejemplo: Entrada: 2024 → Salida: "Es bisiesto"

def es\_bisiesto():

print("Comprobador de años bisiestos")

# Solicitar el año al usuario

try:

año = int(input("Introduce un año: "))

except ValueError:

print("Por favor, introduce un valor numérico válido.")

return

# Determinar si es bisiesto

if (año % 4 == 0 and año % 100 != 0) or (año % 400 == 0):

print(f"{año} es bisiesto")

else:

print(f"{año} no es bisiesto")

# Ejecutar la función

es\_bisiesto()

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

11. Validar contraseñas:

Escribe un programa que solicite una contraseña y valide si es correcta (ejemplo: contraseña fija es 12345). Ejemplo: Entrada: 12345 → Salida: "Acceso concedido".

def validar\_contraseña():

# Contraseña fija

contraseña\_fija = "12345"

print("Validador de contraseña")

# Solicitar contraseña al usuario

contraseña\_ingresada = input("Introduce la contraseña: ")

# Validar contraseña

if contraseña\_ingresada == contraseña\_fija:

print("Acceso concedido")

else:

print("Acceso denegado")

# Ejecutar la función

validar\_contraseña()

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

12. Juego de números:

Genera un número aleatorio entre 1 y 10 y solicita al usuario que adivine el número. Usa if para verificar si acertó o no. Ejemplo: Entrada: 5 → Salida: "¡Felicidades, acertaste!" o "Intenta de nuevo."

import random

def juego\_de\_numeros():

print("¡Bienvenido al Juego de Números!")

# Generar un número aleatorio entre 1 y 10

numero\_secreto = random.randint(1, 10)

try:

# Solicitar un número al usuario

numero\_usuario = int(input("Adivina un número entre 1 y 10: "))

# Verificar si acertó

if numero\_usuario == numero\_secreto:

print("¡Felicidades, acertaste!")

else:

print(f"Intenta de nuevo. El número era {numero\_secreto}.")

except ValueError:

print("Por favor, introduce un número válido.")

# Ejecutar el juego

juego\_de\_numeros()

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

13. Calcular el signo zodiacal:

Solicita el día y mes de nacimiento y determina el signo zodiacal del usuario.Ejemplo: Entrada: 22, marzo → Salida: "Tu signo es Aries".

def obtener\_signo\_zodiaco():

print("Determinador de Signo Zodiacal")

# Solicitar día y mes de nacimiento

try:

dia = int(input("Introduce el día de nacimiento: "))

mes = input("Introduce el mes de nacimiento (ejemplo: marzo): ").lower()

except ValueError:

print("Por favor, introduce un día válido.")

return

# Diccionario para convertir el mes a número

meses = {

"enero": 1, "febrero": 2, "marzo": 3, "abril": 4, "mayo": 5, "junio": 6,

"julio": 7, "agosto": 8, "septiembre": 9, "octubre": 10, "noviembre": 11, "diciembre": 12

}

if mes not in meses:

print("Mes no válido. Asegúrate de escribirlo correctamente.")

return

mes\_num = meses[mes]

# Determinar el signo zodiacal

if (mes\_num == 1 and 20 <= dia <= 31) or (mes\_num == 2 and 1 <= dia <= 18):

signo = "Acuario"

elif (mes\_num == 2 and 19 <= dia <= 29) or (mes\_num == 3 and 1 <= dia <= 20):

signo = "Piscis"

elif (mes\_num == 3 and 21 <= dia <= 31) or (mes\_num == 4 and 1 <= dia <= 19):

signo = "Aries"

elif (mes\_num == 4 and 20 <= dia <= 30) or (mes\_num == 5 and 1 <= dia <= 20):

signo = "Tauro"

elif (mes\_num == 5 and 21 <= dia <= 31) or (mes\_num == 6 and 1 <= dia <= 20):

signo = "Géminis"

elif (mes\_num == 6 and 21 <= dia <= 30) or (mes\_num == 7 and 1 <= dia <= 22):

signo = "Cáncer"

elif (mes\_num == 7 and 23 <= dia <= 31) or (mes\_num == 8 and 1 <= dia <= 22):

signo = "Leo"

elif (mes\_num == 8 and 23 <= dia <= 31) or (mes\_num == 9 and 1 <= dia <= 22):

signo = "Virgo"

elif (mes\_num == 9 and 23 <= dia <= 30) or (mes\_num == 10 and 1 <= dia <= 22):

signo = "Libra"

elif (mes\_num == 10 and 23 <= dia <= 31) or (mes\_num == 11 and 1 <= dia <= 21):

signo = "Escorpio"

elif (mes\_num == 11 and 22 <= dia <= 30) or (mes\_num == 12 and 1 <= dia <= 21):

signo = "Sagitario"

elif (mes\_num == 12 and 22 <= dia <= 31) or (mes\_num == 1 and 1 <= dia <= 19):

signo = "Capricornio"

else:

print("Fecha no válida.")

return

# Mostrar el signo zodiacal

print(f"Tu signo es {signo}.")

# Ejecutar el programa

obtener\_signo\_zodiaco()

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

14. Sistema de calificaciones:

Solicita una calificación numérica y devuelve la letra correspondiente: 90-100: A, 80-89: B, 70-79: C, 60-69: D, Menor a 60: F. Ejemplo: Entrada: 85 → Salida: "Tu calificación es B"

def sistema\_calificaciones():

print("Sistema de Calificaciones")

# Solicitar la calificación numérica

try:

calificacion = float(input("Introduce tu calificación (0-100): "))

except ValueError:

print("Por favor, introduce un valor numérico válido.")

return

# Validar y asignar la calificación en letra

if 90 <= calificacion <= 100:

letra = "A"

elif 80 <= calificacion < 90:

letra = "B"

elif 70 <= calificacion < 80:

letra = "C"

elif 60 <= calificacion < 70:

letra = "D"

elif 0 <= calificacion < 60:

letra = "F"

else:

print("Por favor, introduce una calificación entre 0 y 100.")

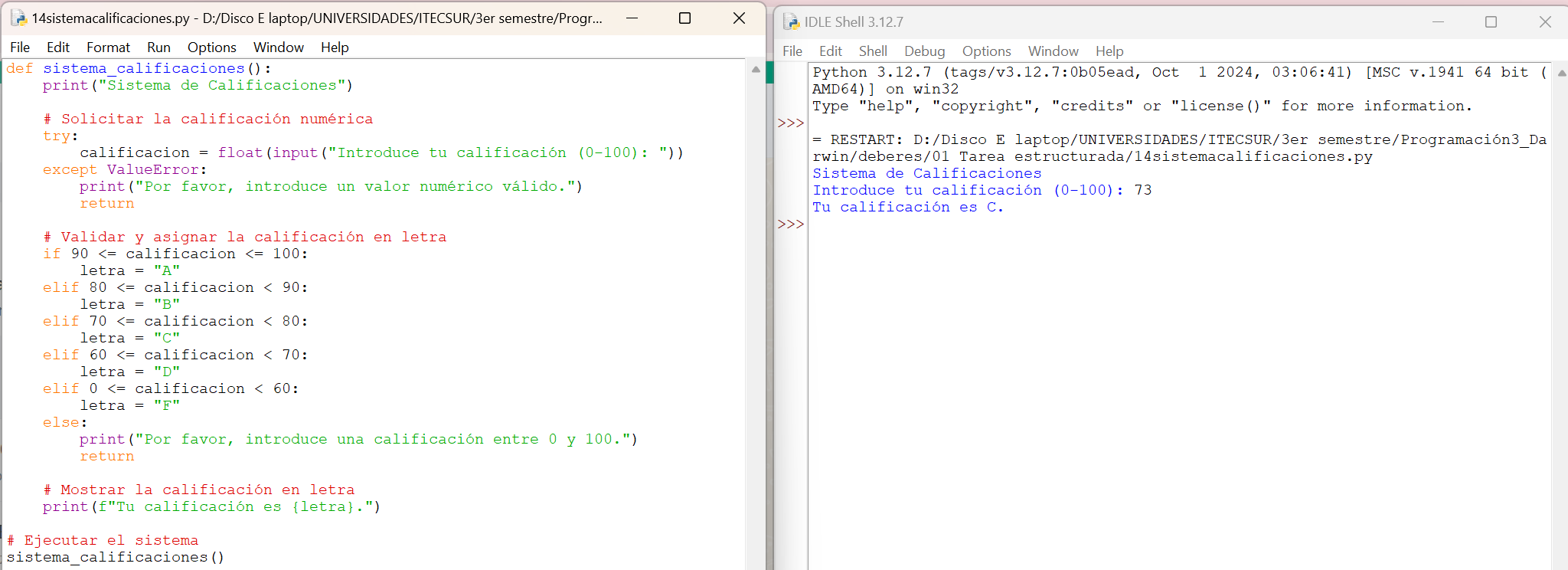
return

# Mostrar la calificación en letra

print(f"Tu calificación es {letra}.")

# Ejecutar el sistema

sistema\_calificaciones()



15. Control de acceso:

Solicita un nombre de usuario y contraseña, y valida si ambos son correctos. Permite tres intentos antes de bloquear el acceso. Ejemplo: Entrada: Usuario: admin, Contraseña: 1234 → Salida: "Bienvenido, admin."

intentos = 3

usuario\_correcto = "admin"

contraseña\_correcta = "1234"

while intentos > 0:

usuario = input("Usuario: ")

contraseña = input("Contraseña: ")

if usuario == usuario\_correcto and contraseña == contraseña\_correcta:

print(f"Bienvenido, {usuario}.")

break

else:

intentos -= 1

print(f"Usuario o contraseña incorrectos. Intentos restantes: {intentos}")

if intentos == 0:

print("Acceso bloqueado.")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

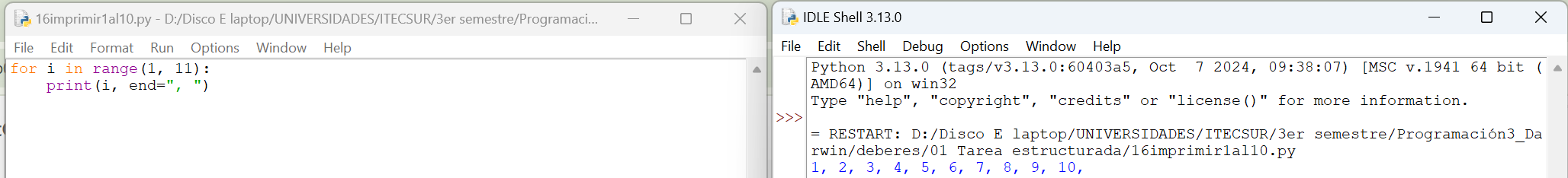
Bucles

16. Imprimir números del 1 al 10:

Usa un bucle para mostrar los números del 1 al 10 en la consola. Salida esperada: 1, 2, 3, ..., 10

for i in range(1, 11):

print(i, end=", ")



17. Suma de los primeros 10 números:

Calcula y muestra la suma de los números del 1 al 10. Salida esperada: 55

suma = sum(range(1, 11))

print(f"La suma de los números del 1 al 10 es: {suma}")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

18. Tabla de multiplicar:

Solicita un número al usuario y muestra su tabla de multiplicar del 1 al 10. Ejemplo: Entrada: 5 → Salida: 5 x 1 = 5, ..., 5 x 10 = 50

numero = int(input("Introduce un número: "))

for i in range(1, 11):

print(f"{numero} x {i} = {numero \* i}")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

19. Contar números pares:

Imprime todos los números pares entre 1 y 20. Salida esperada: 2, 4, 6, ..., 20

for i in range(2, 21, 2):

print(i, end=", ")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

20. Contador regresivo:

Imprime los números del 10 al 1 en orden descendente. Salida esperada: 10, 9, 8, ..., 1

for i in range(10, 0, -1):

print(i, end=", ")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ejercicios intermedios

21. Factorial de un número:

Calcula el factorial de un número ingresado por el usuario (n!). Ejemplo: Entrada: 5 → Salida: 120

numero = int(input("Introduce un número: "))

factorial = 1

for i in range(1, numero + 1):

factorial \*= i

print(f"El factorial de {numero} es: {factorial}")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

22. Números primos:

Encuentra e imprime todos los números primos entre 1 y 50. Salida esperada: 2, 3, 5, 7, ..., 47

for num in range(2, 51):

es\_primo = True

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

es\_primo = False

break

if es\_primo:

print(num, end=", ")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

23. Sumar dígitos:

Solicita un número entero y calcula la suma de sus dígitos. Ejemplo: Entrada: 123 → Salida: 6 (1 + 2 + 3)

numero = int(input("Introduce un número: "))

suma\_digitos = sum(int(digito) for digito in str(numero))

print(f"La suma de los dígitos de {numero} es: {suma\_digitos}")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

24. Invertir un número:

Solicita un número entero y muestra su versión invertida. Ejemplo: Entrada: 1234 → Salida: 4321

numero = int(input("Introduce un número: "))

numero\_invertido = int(str(numero)[::-1])

print(f"El número invertido es: {numero\_invertido}")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

25. Promedio de calificaciones:

Solicita calificaciones al usuario (hasta que ingrese -1) y calcula el promedio. Ejemplo: Entradas: 5, 7, 8, -1 → Salida: Promedio: 6.67

calificaciones = []

while True:

calificacion = float(input("Introduce una calificación (-1 para terminar): "))

if calificacion == -1:

break

calificaciones.append(calificacion)

if calificaciones:

promedio = sum(calificaciones) / len(calificaciones)

print(f"Promedio: {promedio:.2f}")

else:

print("No se ingresaron calificaciones.")

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Funciones

26. Saludo personalizado:

Crea una función que reciba un nombre como parámetro y retorne un saludo. Ejemplo: Entrada: María → Salida: "Hola, María!", en este caso utilice Betty la fea

def saludo(nombre):

return f"Hola, {nombre}!"

print(saludo("Betty la fea"))

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

27. Suma de dos números:

Escribe una función que reciba dos números como parámetros y retorne su suma. Ejemplo: Entrada: 3, 7 → Salida: 10

def suma(a, b):

return a + b

print(suma(3, 7))

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

28. Número par o impar:

Crea una función que reciba un número y retorne True si es par y False si es impar. Ejemplo: Entrada: 4 → Salida: True

def es\_par(numero):

return True if numero % 2 == 0 else False

# Ejemplo de uso

entrada = int(input("Introduce un número: "))

print(f"Entrada: {entrada} → Salida: {es\_par(entrada)}")Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

29. Calcular el cuadrado:

Escribe una función que reciba un número y retorne su cuadrado. Ejemplo: Entrada: 5 → Salida: 25

def calcular\_cuadrado(numero):

return numero \*\* 2

# Ejemplo de uso

entrada = int(input("Introduce un número: "))

print(f"Entrada: {entrada} → Salida: {calcular\_cuadrado(entrada)}")

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

30. Calcular el área de un círculo:

Crea una función que reciba el radio de un círculo y retorne su área. Fórmula: Área = π \* radio^2. Ejemplo: Entrada: 3 → Salida: 28.27 (aproximado)

import math

def area\_circulo(radio):

return round(math.pi \* radio \*\* 2, 2)

print(area\_circulo(3)) # 28.27

**Referencias**

McConnell, S. (2004). Code complete: A practical handbook of software construction (2nd ed.). Microsoft Press.

Van der Veen, J., Tencati, E., & Mann, B. (2017). Learning Python: Learn Python in a day, complete beginners guide to programming. CreateSpace Independent Publishing Platform.

Zhao, Y., Li, X., & Liao, L. (2020). Programming for beginners: Understanding Python and Java. Springer.