

## ACTIVIDAD IA 802

### Actividad 1: Programación en Python

A continuación, se presentan ejercicios para practicar estructuras de control (if, for, while, switch, funciones). Cada ejercicio incluye el código y su explicación para que puedas entender el **por qué** y **para qué** de su uso. Tu tarea es **replicar, modificar y comentar** cada fragmento de código.

#### Estructura condicional if

**Ejercicio 1.** Verifica si un número es positivo, negativo o cero.

```
1  edad = int(input("Introduce tu edad: "))
2
3  if edad < 13:
4      print("Eres un niño.")
5  elif edad < 18:
6      print("Eres un adolescente.")
7  elif edad < 60:
8      print("Eres un adulto.")
9  else:
10     print("Eres un adulto mayor.")
```

#### ¿Por qué if?

La estructura if-elif-else permite clasificar casos mutuamente excluyentes. Aquí usamos rangos para decidir en qué etapa de la vida se encuentra una persona.

#### Preguntas de lógica:

1. ¿Qué sucedería si no incluyéramos el else?

R= Si no incluyéramos el else, cuando el número sea 0, no se mostraría ningún mensaje

2. ¿Cómo cambiarías el código para incluir una categoría “adulto joven” de 18 a 29 años?

```
edad = int(input("Ingresa su edad: "))
```

```
if edad < 18:
```

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

```
elif 18 <= edad <= 29:
```

```
print("Eres un adulto joven")
```

```
elif edad >= 30:
```

```
print("Eres un adulto")
```

3. ¿Qué pasaría si alguien introduce una edad negativa?

R= Se debe validar la entrada para evitar edades negativas

## Ejercicio 2: Validación de contraseña con advertencia

```
1  usuario = input("Usuario: ")
2  password = input("Contraseña: ")
3
4  if usuario == "admin" and password == "secreto123":
5      print("Acceso concedido.")
6  elif usuario != "admin":
7      print("Usuario incorrecto.")
8  else:
9      print("Contraseña incorrecta.")
```

### ¿Por qué if?

Permite validar condiciones combinadas con operadores lógicos (and, or). Aquí se usa para diferenciar fallas de autenticación.

### Preguntas de lógica:

1. ¿Qué operadores lógicos se usan en el código?
2. ¿Por qué es importante validar el usuario antes que la contraseña?
3. ¿Cómo implementarías un intento fallido limitado a 3 veces?

## Ejercicio 3: Mostrar tabla de multiplicar de un número

```

1  numero = int(input("Introduce un número: "))
2
3  for i in range(1, 11):
4      print(f"{numero} x {i} = {numero * i}")
5

```

### ¿Por qué for?

El for se usa cuando conoces de antemano el número de repeticiones. Aquí se usa para imprimir del 1 al 10.

### Preguntas de lógica:

1. ¿Qué hace range(1, 11) exactamente?
2. ¿Cómo podrías modificar el código para imprimir la tabla al revés?
3. ¿Cómo harías que imprima todas las tablas del 1 al 10?

### Ejercicio 4: Contar letras en una palabra (sin espacios)

```

1  palabra = input("Escribe una frase: ")
2  contador = 0
3
4  for letra in palabra:
5      if letra != " ":
6          contador += 1
7
8  print("Número de letras (sin espacios):", contador)
9

```

### ¿Por qué for con if?

El for recorre carácter por carácter, y el if filtra los espacios. Esto combina lógica de control con análisis de datos.

### Preguntas de lógica:

1. ¿Qué otros caracteres podrías querer ignorar además de espacios?
2. ¿Cómo podrías contar solo vocales?
3. ¿Qué pasa si no usas if?

### Ejercicio 5: Validar entrada hasta que sea correcta

```

1  respuesta = ""
2
3  while respuesta != "python":
4      respuesta = input("¿Cuál es el mejor lenguaje de programación?: ")
5
6  print("¡Correcto!")
7

```

### ¿Por qué while?

Se utiliza cuando no sabes cuántas veces se va a repetir el ciclo. El programa sigue preguntando hasta que se cumpla la condición.

### Preguntas de lógica:

1. ¿Qué pasaría si el usuario nunca escribe la palabra correcta?
2. ¿Cómo agregarías un número máximo de intentos?
3. ¿Puedes cambiar el programa para que no importe si escriben "Python" con mayúscula?

### Ejercicio 6: Cajero automático simulado

```

1  saldo = 1000
2  opcion = ""
3
4  while opcion != "salir":
5      print("\n1. Consultar saldo\n2. Retirar\n3. Salir")
6      opcion = input("Elige una opción: ")
7
8      if opcion == "1":
9          print("Saldo actual:", saldo)
10     elif opcion == "2":
11         cantidad = int(input("¿Cuánto deseas retirar?: "))
12         if cantidad <= saldo:
13             saldo -= cantidad
14             print("Retiro exitoso. Nuevo saldo:", saldo)
15         else:
16             print("Fondos insuficientes.")
17     elif opcion == "3":
18         opcion = "salir"
19     else:
20         print("Opción no válida.")
21

```

### ¿Por qué match-case?

Permite manejar múltiples casos claramente, como un switch. Es más legible que varios if.

#### Preguntas de lógica:

1. ¿Qué ventaja tiene match sobre if en este caso?
2. ¿Qué pasa si el usuario pone otro símbolo?
3. ¿Qué deberías validar antes de dividir?

### Ejercicio 7: Menú de operaciones matemáticas

```
1  op = input("Elige una operación (+, -, *, /): ")
2  a = int(input("Primer número: "))
3  b = int(input("Segundo número: "))
4
5  match op:
6      case "+":
7          print("Resultado:", a + b)
8      case "-":
9          print("Resultado:", a - b)
10     case "*":
11         print("Resultado:", a * b)
12     case "/":
13         print("Resultado:", a / b if b != 0 else "No se puede dividir entre cero")
14     case _:
15         print("Operación no válida.")
16
```

### ¿Por qué match-case?

Permite manejar múltiples casos claramente, como un switch. Es más legible que varios if.

#### Preguntas de lógica:

1. ¿Qué ventaja tiene match sobre if en este caso?
2. ¿Qué pasa si el usuario pone otro símbolo?
3. ¿Qué deberías validar antes de dividir?

### Ejercicio 8: Función que devuelve si un número es primo

```
1 def es_primo(n):
2     if n < 2:
3         return False
4     for i in range(2, int(n**0.5) + 1):
5         if n % i == 0:
6             return False
7     return True
8
9 print(es_primo(13)) # True
10
```

#### ¿Por qué función?

Porque encapsulamos una tarea reutilizable. Y se usa la raíz cuadrada como optimización.

#### Preguntas de lógica:

1. ¿Qué hace exactamente el bucle for dentro de la función?
2. ¿Por qué se empieza desde 2 y no desde 1?
3. ¿Qué pasaría si no usáramos return dentro del if?