

## Unidad 2. Desarrollo de algoritmos en Python

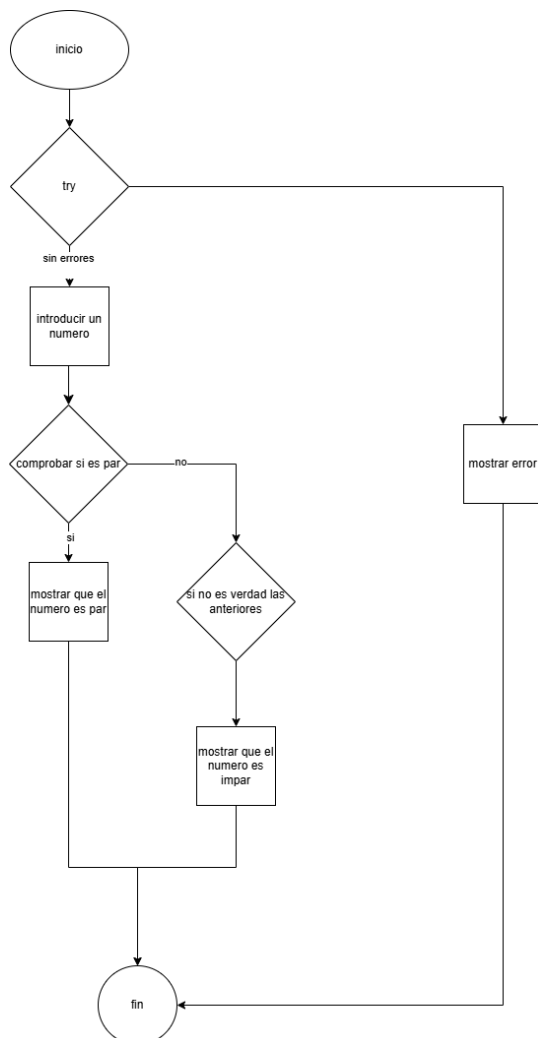
### OPT2 – Tarea individual

- **RA2\_a) Se han creado diagramas de flujo que representen algoritmos de manera efectiva.**

○ **Ejercicio 1.** Realiza un diagrama de flujo que represente un algoritmo que:

1. Pida al usuario un número entero
2. Determine si es par o impar
3. Muestre el resultado en pantalla

El diagrama debe contener claramente el inicio, el proceso de decisión y el fin.



- **RA2\_b) Se ha escrito código en Python que implemente algoritmos diseñados.**

- **Ejercicio 2.** Escribe un programa en Python que implemente el algoritmo del ejercicio anterior.

try:

```
num = int(input("Ingrese un número entero: "))
```

```
if num % 2 == 0:
```

```
    print("El número ",num," es par.")
```

```
else:
```

```
    print("El número", num,"es impar.")
```

except ValueError:

```
    print("Error: Entrada no válida. Por favor, ingrese un número entero.")
```

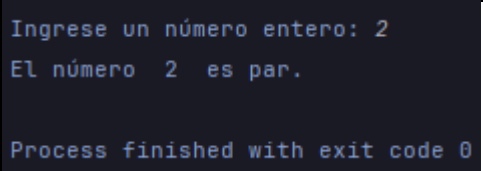
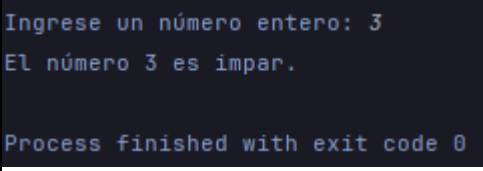
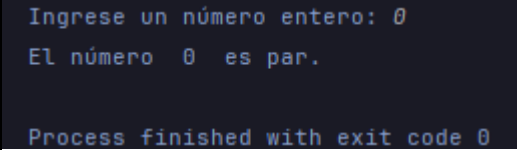
- **RA2\_c) Se han realizado pruebas para validar la funcionalidad de los algoritmos.**

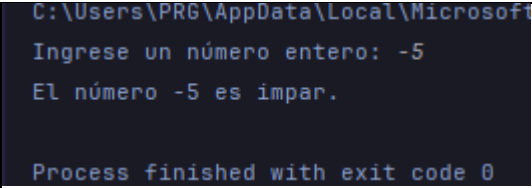
- **Ejercicio 3.** Realiza tres pruebas diferentes con tu programa del ejercicio 2:

- Un número par
- Un número impar
- Un número negativo o 0

Presenta las pruebas en una tabla con:

- Número introducido
- Resultado esperado
- Resultado obtenido
- ¿Coincide? ✓ / ✗

Número introducido	Resultado esperado	Resultado obtenido	¿Coincide?
2	El número 2 es par		✓
3	El número 3 es impar		✓
0	El número 0 es par		✓

-5	El número -5 es <u>impar</u>		✓
----	------------------------------	--	---

• **RA2\_d) Se han identificado y solucionado errores en algoritmos.**

- **Ejercicio 4.** Analiza el siguiente código erróneo que quiere hacer lo mismo que el ejercicio 2:

```
num = input("Introduce un número: ")
if num % 2 = 0:
    print("Es par")
else
```

- 1. Identifica al menos dos errores presentes en el código
- 2. Escribe la versión corregida
- 3. Ejecuta el programa corregido y adjunta captura

```
'''
version: erronea.
num = input("Introduce un número: ")
if num % 2 = 0:
    print("Es par")
else
    print("Es impar")
'''
```

# Versión corregida:

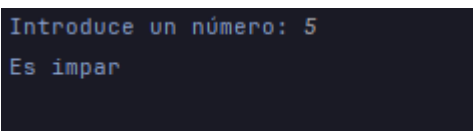
#buena practica para manejar errores de entrada

try:

```
    num = int(input("Introduce un número: "))
    # uso correcto del operador de comparación
    if num % 2 == 0:
        # uso adecuado de la tabulación
        print("Es par")
    else:
```

except ValueError:

```
    print("Error: Entrada no válida. Por favor, ingrese un número entero.")
```



- **RA2\_e) Se ha explicado el proceso de desarrollo de un algoritmo de forma clara.**

- **Ejercicio 5.** Párate a pensarlo un momento y explica, con tus palabras:

- **Cómo ideaste el algoritmo**

Primero pensé en cómo solucionar el problema donde, primero tenía que haber algún dato que comprobar si era par, después pensé en cómo se comprobaba si era para o impar así que elegí un `if` que permite comprobar una condición y si es `true` escoge el bloque de código y si es `false` avanza sin ejecutar el bloque y después simplemente dentro del bloque mostré que era y ya por ultimo pensando en errores u optimizarlo porque si no nos bajas puntos pensé en poner un `try except`, para manejar errores.

- **Por qué funciona correctamente**

Funciona correctamente porque los pasos a seguir son los adecuados y por qué se le han añadido extras que impiden errores típicos como el `try` y `catch`.

- **Qué pasos serían necesarios para adaptarlo, por ejemplo, para detectar si un número es múltiplo de 3**

Solo tendrías que hacer tres modificaciones, cambiar la condición del `if` por `num%3 == 0`, el primer `print` tendrías que poner que (num, " es múltiplo de tres ") y en el segundo (num, " no es múltiplo de tres ")

Extensión recomendada: entre 6 y 10 líneas.

La entrega de la tarea constará en un documento **en formato PDF**

