



cristiandev /
PythonML



<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights



PythonML / TP1_Comisión_4_para_entregar.ipynb



cristian 4 minutes ago



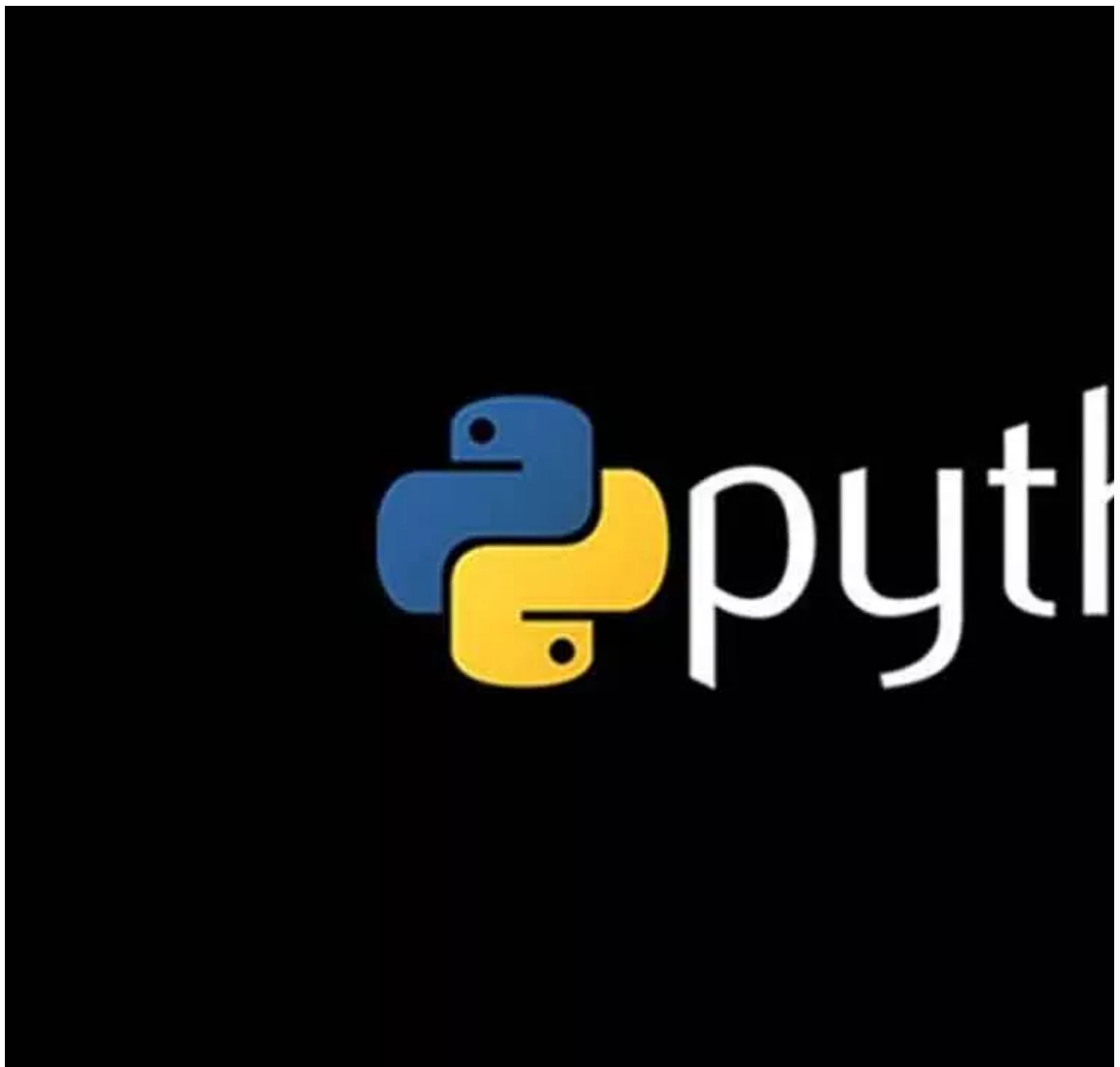
2.4 MB





Trabajo Practico 1

Introduction al Desarrollo en Python



Indice

[Ejercicio 1](#) - Operadores


[Ejercicio 2](#) - Estructura de control condicional

[Ejercicio 3](#) - Ciclo for

[Ejercicio 4](#) - Rango

[Ejercicio 5](#) - Ciclo while y funciones

Símbolos en un Diagrama de Flujo

Nombre	Función	Símbolo
Terminal	Determina el inicio o final de un programa.	

Proceso, acción	Representa cualquier operación que se le aplique a la información guardada. Esta operación debe cambiar la posición, valor o forma de la información a la que se le aplica. Comúnmente son operaciones matemáticas.	
Entrada / Salida	Nos indica cualquier tipo de dato que ingrese o salga del sistema.	
Decisión	Representa una instrucción de comparación lógica entre dos o más valores. Si el valor de la comparación es verdadero, el programa toma una dirección, si es falsa, toma otra.	
Línea de flujo	Estas indican la dirección y el orden en el que se ejecuta el algoritmo.	
Impresora	Indica la salida de un dato o expresión, normalmente en formato visual.	
Pantalla	Indica la salida de un dato, normalmente información que se muestra en formato visual.	
Entrada	Símbolo alternativo para simbolizar la entrada de datos.	
Teclado, entrada manual	Se utiliza para representar la entrada manual de datos por parte del usuario.	
Conector misma página	Se utiliza para conectar o indicar la continuación de un algoritmo en una misma página.	
Conector diferente página	Se utiliza para conectar o indicar la continuación de un algoritmo en una página distinta.	

Fuente: https://solucioningenieril.com/imagenes/asignaturas/programacion_en_c/tema_4/1.png





EJERCICIO 1 ESTRUCTURA DE CONTROL CONDICIONAL

[Ir al índice](#)

Determinar si un numero ingresado por el usuario es PAR

Solución 1.1

Usando el operador módulo (%) que devuelve el resto de la división de dos números:

- Si el resto es 0, el número es par.
- Si el resto es 1, el número es impar.

```
In [ ]: #Por ejemplo
        resto=5%2
        print(resto)
```

```
In [ ]: # Porque al dividir 5 entre 2, el resto es 1.
```

```
In [ ]: # Paso 1: Pido un número al usuario y lo convierto a número entero
        numero=input("Ingrese un número: ")
        print(f"Lo que el usuario ingresa siempre es una cadena: {numero}")
        # numero_entero=int(numero)
        # print(f"Convertido a número entero: {numero_entero}")
```

```
In [ ]: # Paso 2: Verifico si lo que el usuario ingresa es un número entero
        cadena='a'
        if cadena.isnumeric():
            print(f'El usuario ingresó un número: {cadena}')
        else:
            print(f'{cadena} no es un número')
```

```
In [ ]: """
        cadena=1
        if cadena.isnumeric():
            print(f'El usuario ingresó un número: {cadena}')
        else:
            print(f'{cadena} no es un número')
        """
        # AttributeError: 'int' object has no attribute 'isnumeric'
        # Como cadena es un objeto 'int' no tiene un atributo 'isnumeric'
```

Uso la función **str()** para convertir la cadena a un objeto str antes de usar el método **isnumeric()**

De esta forma, nos aseguramos que el método **isnumeric()** se aplica sobre una cadena de texto y no sobre

De esta forma, nos aseguramos que el método `isnumeric()` se aplica sobre una cadena de texto, y no sobre

```
In [ ]:
cadena=1
if str(cadena).isnumeric():
    print(f'El usuario ingresó un número entero: {cadena}')
else:
    print(f'{cadena} no es un número entero')
```

```
In [ ]:
cadena=1.5
if str(cadena).isnumeric():
    print(f'El usuario ingresó un número entero: {cadena}')
else:
    print(f'{cadena} no es un número entero')
```

Uso la función **isdecimal()**:

- Devuelve **True** si la cadena contiene solo dígitos (caracteres del 0 al 9)
- Devuelve **False** en caso contrario.

```
In [ ]:
cadena=1.5

print('Es un número entero? ',str(cadena).isnumeric())
```

```
In [ ]:
cadena=1.5

print('Es un número decimal?', str(cadena).isdecimal())
```

```
In [ ]:
```

Uso la función **float()** para intentar convertir la cadena a un número de punto flotante.

- Si la conversión tiene éxito, significa que la cadena es un número decimal válido.
- Si la conversión falla, significa que la cadena no es un número decimal válido.

```
In [ ]:
cadena = "1.5"
try:
    numero = float(cadena)
    print(f"El usuario ingresó un número decimal: {cadena}")
except:
    print(f"{cadena} no es un número decimal")

# try significa intentar ejecutar algo
# except se ejecuta en caso de fallar el intento
```

Uso la función **int()** convierte un valor especificado en un número entero.

- Devuelve un objeto entero construido a partir de un número o una cadena x
- Devuelve 0 si no se dan argumentos.

```
In [ ]:
# Convertir un número de coma flotante en un entero
print(int(3.14))
```

```
Out[ ]: 3
```

```
In [ ]:
# Convertir una cadena en un entero
print(int("42"))
```

```
In [ ]: # Convertir una cadena hexadecimal en un entero
        print(int("0x2a", 16))
```

42

```
In [ ]: # Devolver 0 si no se dan argumentos
        print(int())
```

0

```
In [ ]: # Convertir una cadena no válida en un entero
        # print(int("abc"))

        # ValueError: invalid literal for int() with base 10: 'abc'
        # Dice que 'abc' no es un número entero
```

```
In [ ]: cadena = "abc"
        try:
            numero = int(cadena)
            print(f"El usuario ingresó: {cadena}")
            print(f"Convertido a número entero es: {numero}")
        except:
            print(f"{cadena} no es un número entero")

        # try significa intentar ejecutar algo
        # except se ejecuta en caso de fallar el intento
```

abc no es un número entero

1 - GANÁ MILLAS PARA EL DESTINO PYTHON

¿Cómo vamos a validar si lo que ingresa el usuario por teclado es un número?

Responde "A es mejor" o "B es mejor":

Tus comentarios (opcional): A es mejor

Explicacion: Ya que nos permite, a pesar del error, continuar ejecutando el código. Es decir el programa no

```
In [ ]: """
        OPCION A

        cadena=input("Ingrese un número entero:")
        try:
            numero = int(cadena)
            print(f"El usuario ingresó: {cadena}")
            print(f"Convertido a número entero es: {numero}")
        except:
            print(f"{cadena} no es un número entero")
        """
```

```
In [ ]: """
        OPCION B

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

        """
```

Solución 1.2

Determinamos si un número dado es par

```
In [ ]: # Pedimos un número al usuario
```

```

cadena=input("Ingrese un número entero: ")

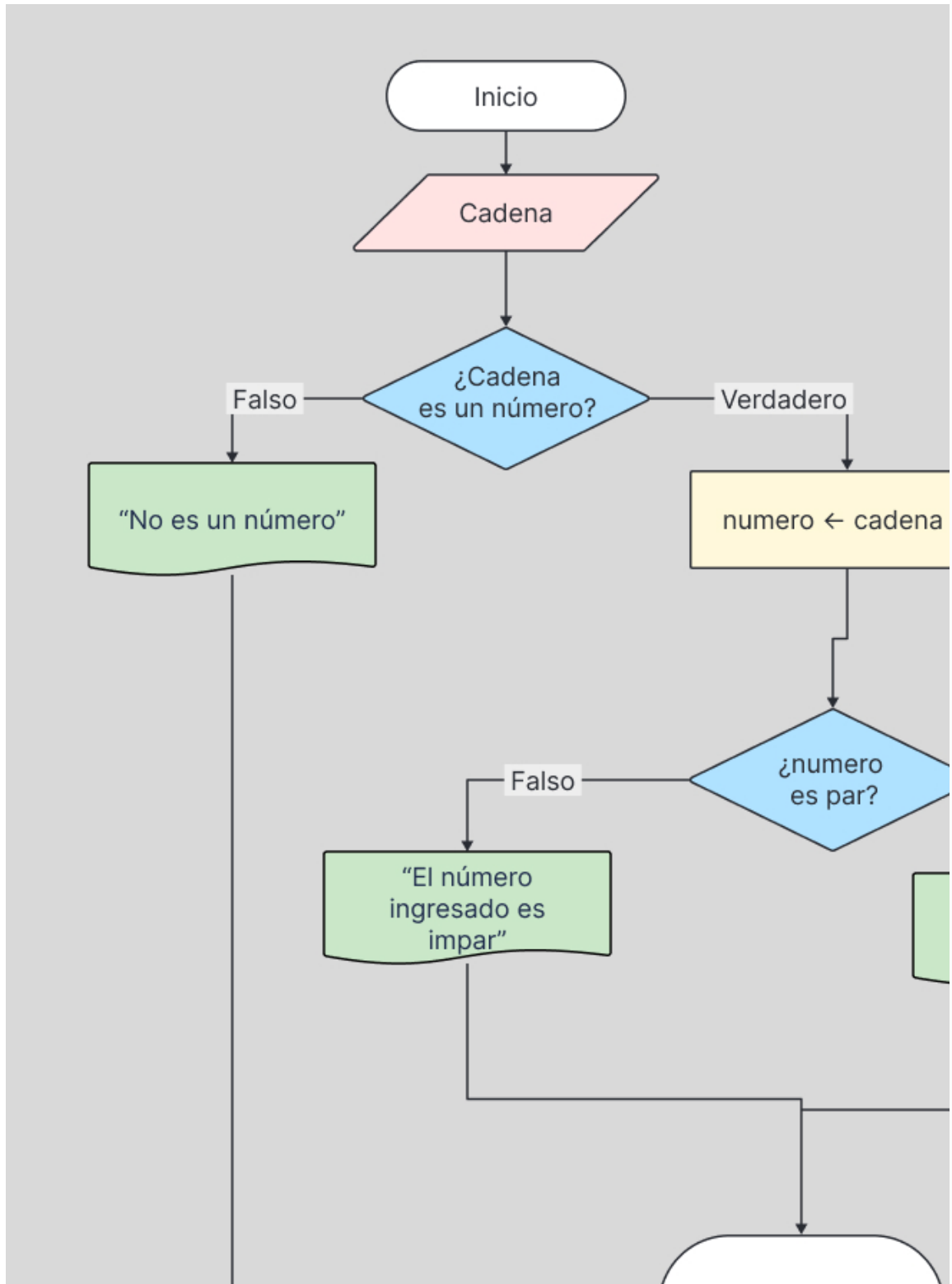
try:
    numero = int(cadena)
    print(f"Lo que ingresa por teclado: {cadena}")
    print(f"Convertido a número entero es: {numero}")

    # Vemos si el número es par o impar usando el operador %
    if numero % 2 == 0:
        print("El número", numero, "es par.")
    else:
        print("El número", numero, "es impar.")

except:
    print(f"{cadena} no es un número entero")

```

Ingrese un número entero: asdasd
 asdasd no es un número entero







EJERCICIO 2

[Ir al índice](#)

Determinar si un numero ingresado por el usuario esta DENTRO DEL RANGO [1,10] y es par. Si el número n RANGO'.

```
In [ ]: # El rango

rango=range(1,11)

print(rango)
```

```
range(1, 11)
```

```
In [ ]: # Veamos Los números individuales del rango

for numero in rango:
    print(numero)
```

```
1
2
3
4
5
6
7
8
9
10
```

```
In [ ]: # Convertimos el rango en una lista
```

```
""" Convertimos el rango en una lista """
```

```
lista=list(rango)
```

```
print(lista)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [ ]:
```

```
# Veamos si un número está en el rango del 1 al 10
# Y además, si el número es par.

numero=5

if numero in range(1, 11) and numero % 2 == 0:
    print("El número", numero, "está dentro del rango [1, 10] y es par.")
else:
    print("Número fuera del rango o es número impar.")
```

```
In [1]:
```

```
# Veamos si un número ni está en el rango ni es impar
# Nos preguntamos si es impar, mostramos "es impar".

# numero = 5
# if numero>=1 and numero<=10 and (numero)%2==0:
#     print("El número está dentro del rango y es par.")
# else:
#     if (numero)%2==1:
#         if numero>=1 and numero<=10:
#             print("Número dentro del rango pero impar")
#         else:
#             print("Número fuera del rango y es impar")
#     else:
#         print("El numero es par pero está fuera del rango")

# Aqui modificare un poco el codigo
numero = 5

if (numero>=1 and numero<=10) and (numero)%2==0:
    print("El número está dentro del rango y es par.")
elif numero>=1 and numero<=10:
    print("Número dentro del rango pero impar")
elif numero % 2 == 0:
    print("El numero es par pero está fuera del rango")
else:
    print("Número fuera del rango y es impar")
```

Número dentro del rango pero impar

2 - GANÁ MILLAS PARA TU DESTINO PYTHON

¿Es cierto que ambos algoritmos hacen lo mismo?

Responde "VERDADERO, ambas opciones hacen lo mismo" o "FALSO, no hacen lo mismo":

Tus comentarios (opcional): VERDADERO, ambas opciones hacen lo mismo

Explicación: Primero me genero una duda al ver que en el 2do algoritmo realiza un poco diferente las preguntas: ¿el número está dentro del rango y si es o no par.

```
In [ ]:
```

```
"""
OPCION A

cadena=input("Ingrese un número entero:")
try:
    numero=int(cadena)
    print(f"El usuario ingresó: {cadena}")
    print(f"Convertido a número entero es: {numero}")
```

```
if numero>=1 and numero<=10 and (numero)%2==0:
    print("El número está dentro del rango y es par.")
else:
    if (numero)%2==1:
        if numero>=1 and numero<=10:
            print("Número dentro del rango pero impar")
        else:
            print("Número fuera del rango y es impar")
    else:
        print("El numero es par pero está fuera del rango")
except:
    print(f"{cadena} no es un número entero")

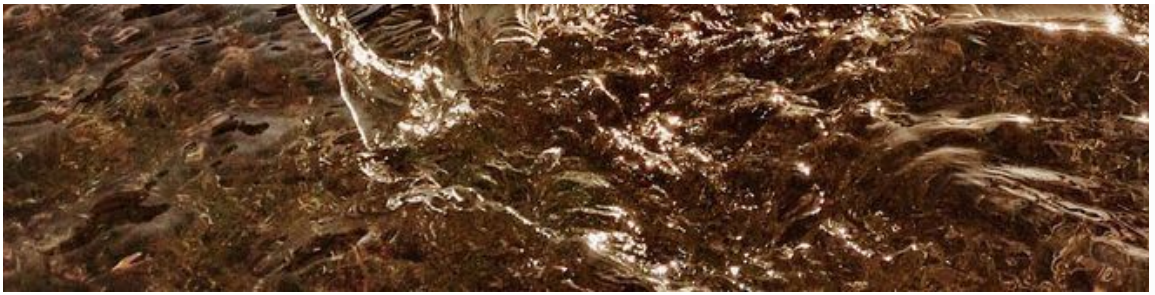
"""
```

In []:

```
"""
OPCION B

cadena=input("Ingrese un número entero:")
try:
    numero=int(cadena)
    print(f"El usuario ingresó: {cadena}")
    print(f"Convertido a número entero es: {numero}")
    if numero in range(1, 11) and numero % 2 == 0:
        print("El número está dentro del rango y es par.")
    elif numero in range(1, 11) and numero % 2 == 1:
        print("Número dentro del rango pero impar")
    elif numero % 2 == 0:
        print("El número es par pero está fuera del rango")
    else:
        print("Número fuera del rango y es impar")
except:
    print(f"{cadena} no es un número entero")
"""
```





EJERCICIO 3 CICLO FOR

[Ir al índice](#)

3 - Ganá millas para tu destino Python

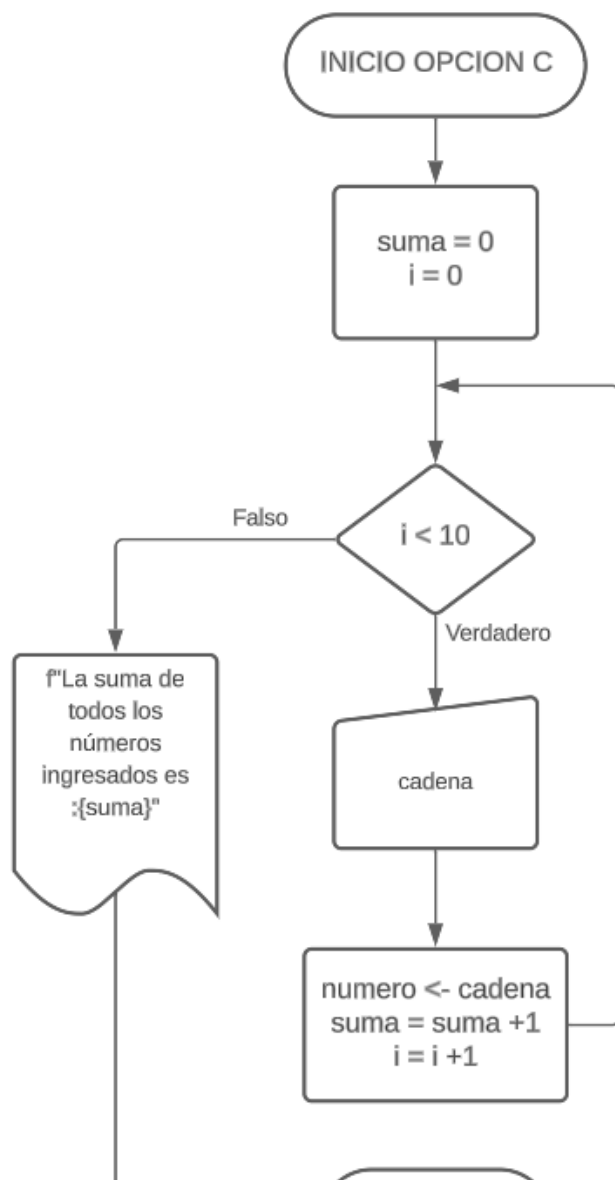
Este algoritmo solicita al usuario que ingrese 10 números enteros y calcula la suma de todos los números e

Responde cuál es la opción correcta:

Tus comentarios (opcional): C es la opcion correcta

Explicación: Dado que i recorre un rango de 10 numeros (0..9) e inicializa el acumulador llamado suma fuer

Ganás millas opcionales extra, si te animás a hacer el diagrama de flujo, con la opción que consideras corre





```
In [ ]: """
OPCION A

suma=0
for i in range(1,10):
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma de todos los números ingresados es :{suma}")

"""
```

```
In [ ]: """
OPCION B

for i in range(10):
    suma=0
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma total es: {suma}")

"""
```

```
In [ ]: """
OPCION C

suma=0
for i in range(10):
    cadena = input(f"{i} - Ingresá un número entero: ")
    try:
        numero = int(cadena)
        print(f"El usuario ingresó: {cadena}")
        print(f"Convertido a número entero es: {numero}")
        suma+=numero
    except:
        print(f"{cadena} no es un número entero")
print(f"La suma de todos los números ingresados es :{suma}")

"""
```





EJERCICIOS 4

[Ir al índice](#)

```
In [ ]: # Paso 1: Mostrar todos Los números enteros en el rango del 56 al 1230.

for numero in range(56, 1231):
    print(numero)
```

```
In [6]: # Paso 2: Pedir al usuario que ingrese el valor a.
cadena_1 = input('Ingresar el numero a: ')
```

```
In [7]: # Paso 3: Pedir al usuario que ingrese el valor b.
cadena_2 = input('Ingresar el numero a: ')
```

```
In [8]: # Paso 4: Mostrar todos Los números enteros en el rango desde a hasta b.
try:
    a = int(cadena_1)
    b = int(cadena_2)
    # Controlamos que a <= b
    if a > b:
        aux = b
        b = a
        a = aux
```

```
for i in range(a,b+1):  
    print(i)  
except:  
    print('Error al convertir a entero')
```

315
316
317
318
319
320
321

4 - Ganá millas para tu destino Python

Completá el código para el paso 2, paso 3 y paso 4.

Tus comentarios (opcional): Al principio pido los numeros pero no los convierto por el tema de controlar lo me muestre el respectivo mensaje y de lo contrario entonces me mostrara los enteros desde a hasta b. Para cumpla siempre, dado que si $a > b$ entonces en el intervalo $[a, b]$ no habrá ningun valor. La solucion que le





EJERCICIO 5 CICLO WHILE

[Ir al índice](#)

Solicitar al usuario que ingrese un número, si es negativo o nulo , solicitar reiteradamente hasta que ingrese

5 - Ganá millas para tu destino Python

Mirá este desarrollo paso a paso, y completá en cada parte del código donde dice completar, reemplazá la

Tus comentarios (opcional):

```
In [ ]: # Paso 1: Dado un número entero, veamos si es positivo, en caso contrario, es negativo o cero.

numero=2
if numero > 0:
    print(f'El número {numero} es positivo.')
else:
    print(f'El número {numero} es negativo o cero.')
```

El número 2 es positivo o cero.

```
In [ ]: # Paso 2: Definimos una función positivo() que recibe un número entero
#         Devuelve 1 si la entrada es positivo.
#         Devuelve 0 si la entrada es un número negativo o cero.

def positivo(numero_entrada):
    if numero_entrada > 0:
        salida=1
    else:
        salida=0
    return salida

print(positivo(-5))
```

0

```
In [9]: # Paso 3: Modificamos la función anterior para que en vez de un número,
#         recibe una cadena como parámetro de entrada.
#         Intentamos ver si esa cadena es un número, antes de ver si es positivo.
#         Si el número es positivo, la función devuelve 1.
#         Si no es un número, o bien, es negativo o es cero,
#         La función devuelve 0.

def positivo(cadena):
    try:
        numero_entrada = int(cadena) # Completar aquí con una función que convierta cadena a nú
        if numero_entrada > 0:
            salida=1
        else:
            salida=0
    except:
        salida=0
    return salida

print(positivo("a"))
```

0

```
In [11]: # Paso 4: Solicitamos al usuario que ingrese un número entero
#         Llamamos a la función positivo para que nos devuelva
#         1 si es un número entero positivo
#         0 si es negativo o cero o no es un número.
#         Recuerda que la función que quedó definida en el paso anterior,
#         No hace falta volver a definirla, La podemos usar en este paso.

algo=input("Ingrese un número entero: ")

espositivo=positivo(algo) # Completar aquí con la llamada a la función

if espositivo:
    print(f"El usuario ingresó un número positivo: {algo}")
else:
    print(f"El usuario ingresó algo que no es un número positivo.")
```

El usuario ingresó algo que no es un número positivo.

```
In [ ]: # Paso 5: Veamos cómo funciona el ciclo while

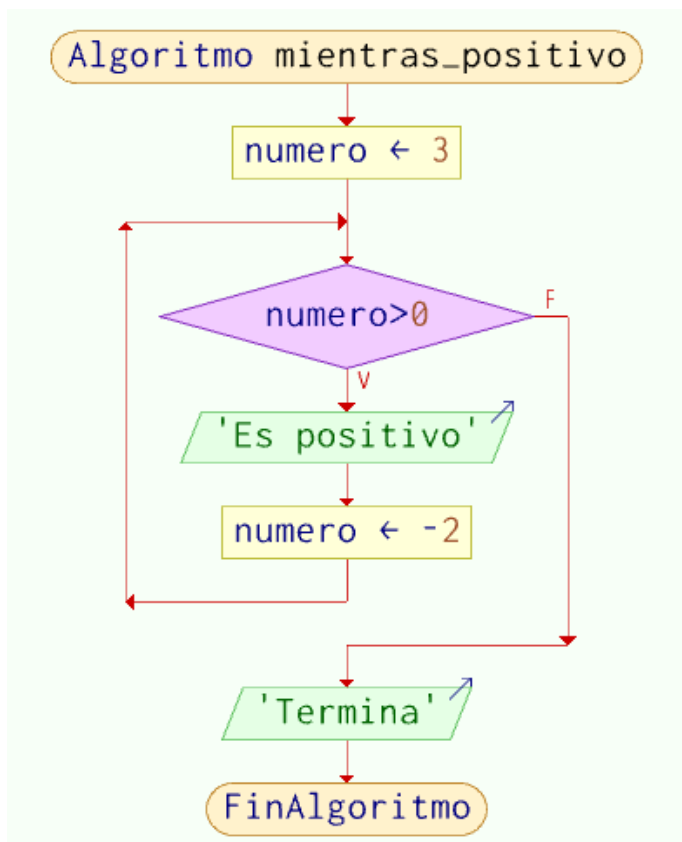
numero=3

while numero > 0:
    print("es positivo")
    numero=-2

print("El ciclo while termina porque el segundo número es negativo.")
```

es positivo

El ciclo while termina porque el segundo número es negativo.



```
In [2]: # Paso 6: Pasamos en limpio lo que desarrollamos hasta ahora
```