



# Recuperación Iso

---

Cristian Oliva Llanos

Implantación de sistemas operativos

ASIR

10/11/2023



## ÍNDICE

<b>1. PROGRAMACIÓN PYTHON.....</b>	<b>3</b>
1.1 Código 1.....	3
1.2 Código 2.....	5
1.3 Código 3.....	7
1.4 Enunciado 1.....	9
1.5 Enunciado 2.....	9
1.6 Enunciado 3.....	9
1.7 Enunciado 4.....	9
1.8 Enunciado 5.....	9
<b>2. LINUX.....</b>	<b>10</b>
2.1 Supuesto práctico.....	10
2.2 Comandos LINUX.....	10
<b>3. WINDOWS.....</b>	<b>10</b>
3.1 Supuesto práctico.....	10
<b>4. GESTIÓN DE RECURSOS Y OTROS.....</b>	<b>10</b>
4.1 Comandos para la gestión de procesos y memoria.....	10
4.2 Licencias 1.....	10
4.3 Licencias 2.....	10

# 1. PROGRAMACIÓN PYTHON

## 1.1 Código 1

Realiza un programa en el que pida al usuario tres números distintos y de los tres que muestre el mayor, para finalizar el programa presione la tecla 0.

### Implantación y solución en python

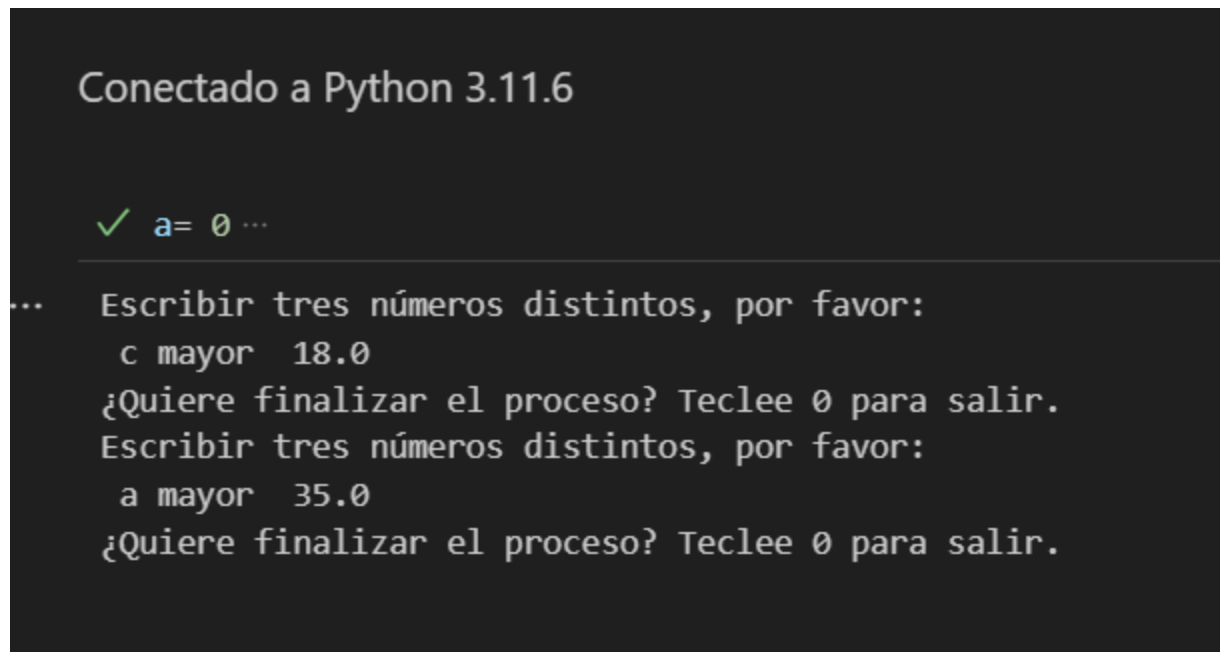
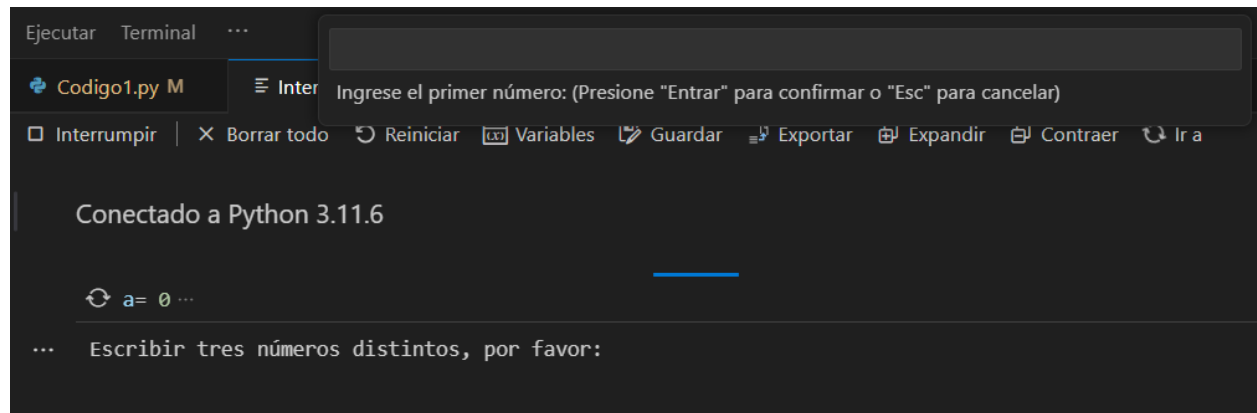
Código dado para realizar el ejercicio.

```
a= 0
b= 0
c= 0
final= 1

while final != 0:
    print("Escribir tres números distintos, por favor: ")
    a= float(input("Ingrese el primer número: "))
    b= float(input("Ingrese el segundo número: "))
    c= float(input("Ingrese el tercer número: "))

    if a>b:
        if a>c:
            print(" a mayor ", a)
        else:
            print(" c mayor ", c)
    else:
        if b>c:
            print(" b mayor ", b)
        else:
            print(" c mayor ", c)

    print( "¿Quiere finalizar el proceso? Teclee 0 para salir.")
    final= int(input())
```



## 1.2 Código 2

Realice un programa en el que se le pregunte al usuario si quiere realizar o no el programa si lo quiere realizar realice un bucle en el que se le vaya pidiendo al usuario números y al llegar a 10 números salga del bucle.

### Implantación y solución en python

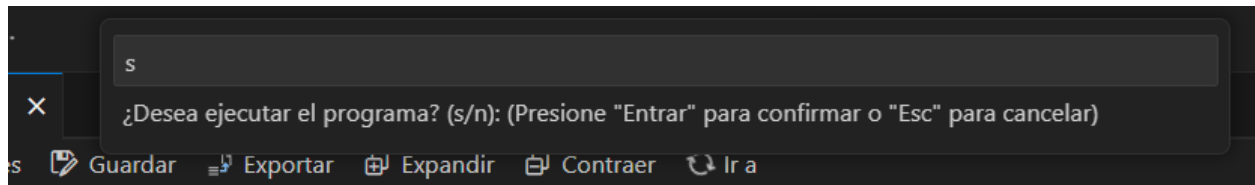
Código dado para realizar el ejercicio.

```
def un_proceso():
    a= 0
    r= 0
    c= 0
    pr= input("¿Desea ejecutar el programa? (s/n): ")

    if pr == "s":
        c = 0
        while c < 10:
            a = int(input("Ingrese un número: "))
            c += 1
            r = a % 2
            if r == 0:
                print(str(a) + 'seleccionado')
            else:
                print(str(a) + 'no seleccionado')
            print(r)
        print("ya terminamos")

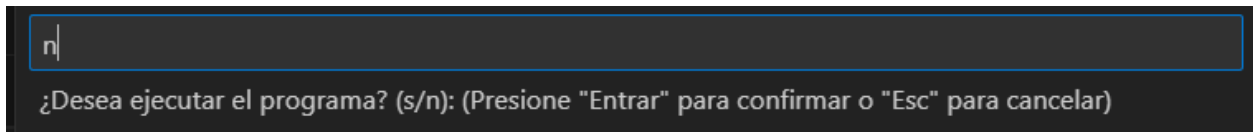
un_proceso()
```

Si el usuario quiere realizar el programa.

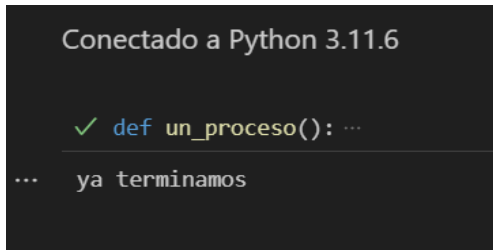


```
1 no seleccionado
1
2 seleccionado
0
3 no seleccionado
1
4 seleccionado
0
5 no seleccionado
1
6 seleccionado
0
7 no seleccionado
1
8 seleccionado
0
9 no seleccionado
1
10 seleccionado
0
ya terminamos
```

Si el usuario no quiere realizar el programa.



```
n|
¿Desea ejecutar el programa? (s/n): (Presione "Entrar" para confirmar o "Esc" para cancelar)
```



```
Conectado a Python 3.11.6

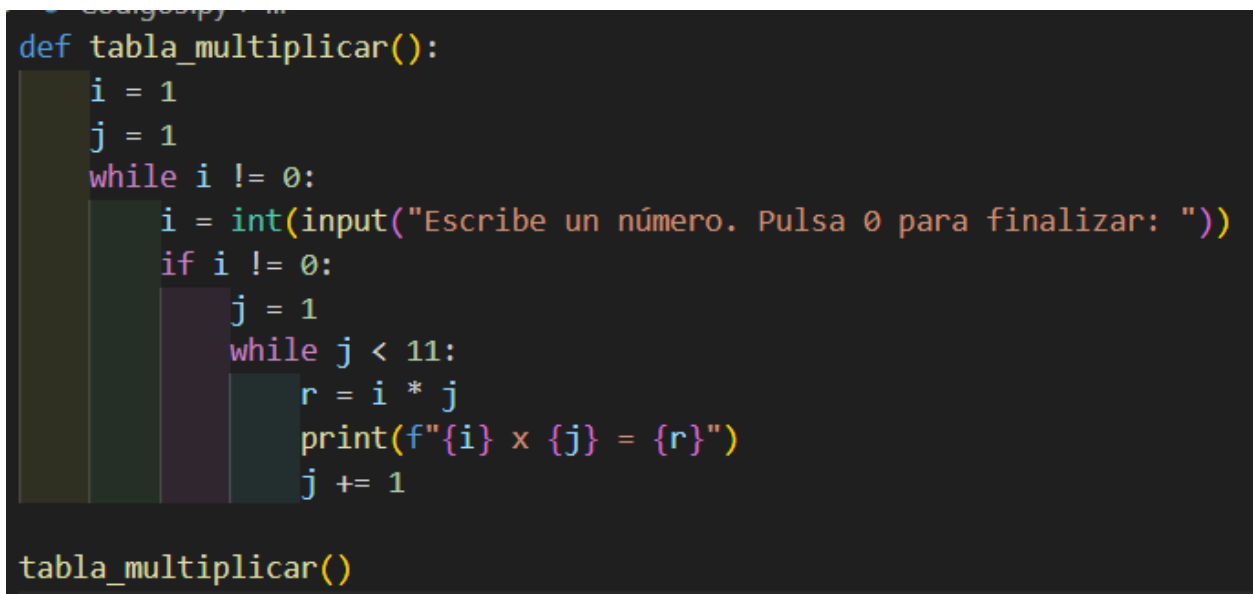
✓ def un_proceso(): ...
... ya terminamos
```

### 1.3 Código 3

Realiza un proceso en el que te muestre la tabla de multiplicar y para finalizar pulse la tecla 0.

#### Implantación y solución en python

Código dado para realizar el ejercicio.



```
def tabla_multiplicar():
    i = 1
    j = 1
    while i != 0:
        i = int(input("Escribe un número. Pulsa 0 para finalizar: "))
        if i != 0:
            j = 1
            while j < 11:
                r = i * j
                print(f"{i} x {j} = {r}")
                j += 1
            i = 0
    tabla_multiplicar()
```

8

Escribe un número. Pulsa 0 para finalizar: (Presione "Entrar" para confirmar o "Esc" para cancelar)

5) Reiniciar Variables Guardar Exportar Expandir Contraseña

0

Escribe un número. Pulsa 0 para finalizar: (Presione "Entrar" para confirmar o "Esc" para cancelar)

Conectado a Python 3.11.6

✓ `def tabla_multiplicar():`

```
.. 8 x 1 = 8
    8 x 2 = 16
    8 x 3 = 24
    8 x 4 = 32
    8 x 5 = 40
    8 x 6 = 48
    8 x 7 = 56
    8 x 8 = 64
    8 x 9 = 72
    8 x 10 = 80
```





## **1.4 Enunciado 1**

[https://github.com/colival03/Oliva\\_Cristian\\_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado1.py](https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado1.py)

## **1.5 Enunciado 2**

[https://github.com/colival03/Oliva\\_Cristian\\_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado2.py](https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado2.py)

## **1.6 Enunciado 3**

[https://github.com/colival03/Oliva\\_Cristian\\_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado3.py](https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado3.py)

## **1.7 Enunciado 4**

[https://github.com/colival03/Oliva\\_Cristian\\_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado4.py](https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado4.py)

## **1.8 Enunciado 5**

[https://github.com/colival03/Oliva\\_Cristian\\_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado5.py](https://github.com/colival03/Oliva_Cristian_ReclSO/blob/ea1ce8d8e16f9474b7fe5ccabd589984b4296665/Python/Enunciado5.py)



## **2. LINUX**

### **2.1 Supuesto práctico**

### **2.2 Comandos LINUX**

## **3. WINDOWS**

### **3.1 Supuesto práctico**

## **4. GESTIÓN DE RECURSOS Y OTROS**

### **4.1 Comandos para la gestión de procesos y memoria**

### **4.2 Licencias 1**

### **4.3 Licencias 2**