



**CICLO: [DAW/DAM]  
MÓDULO DE [PYTHON]**

## **[Tarea Nº3]**

**Alumno:  
[Xavi Morro Sanz]  
[41514746E]**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenece a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

## Contenido

- |  |   |
|--|---|
| 1. Documentos que se adjuntan a este informe .....                     | 2 |
| 2. Resto de epígrafes que componen el desarrollo de este informe. .... | 2 |

(Una vez realizado el informe, no olvidar actualizar esta tabla del índice (**F9 + Actualizar toda la tabla**), con el fin de que se actualicen todos los epígrafes y números de página)

## 1. Documentos que se adjuntan a este informe.

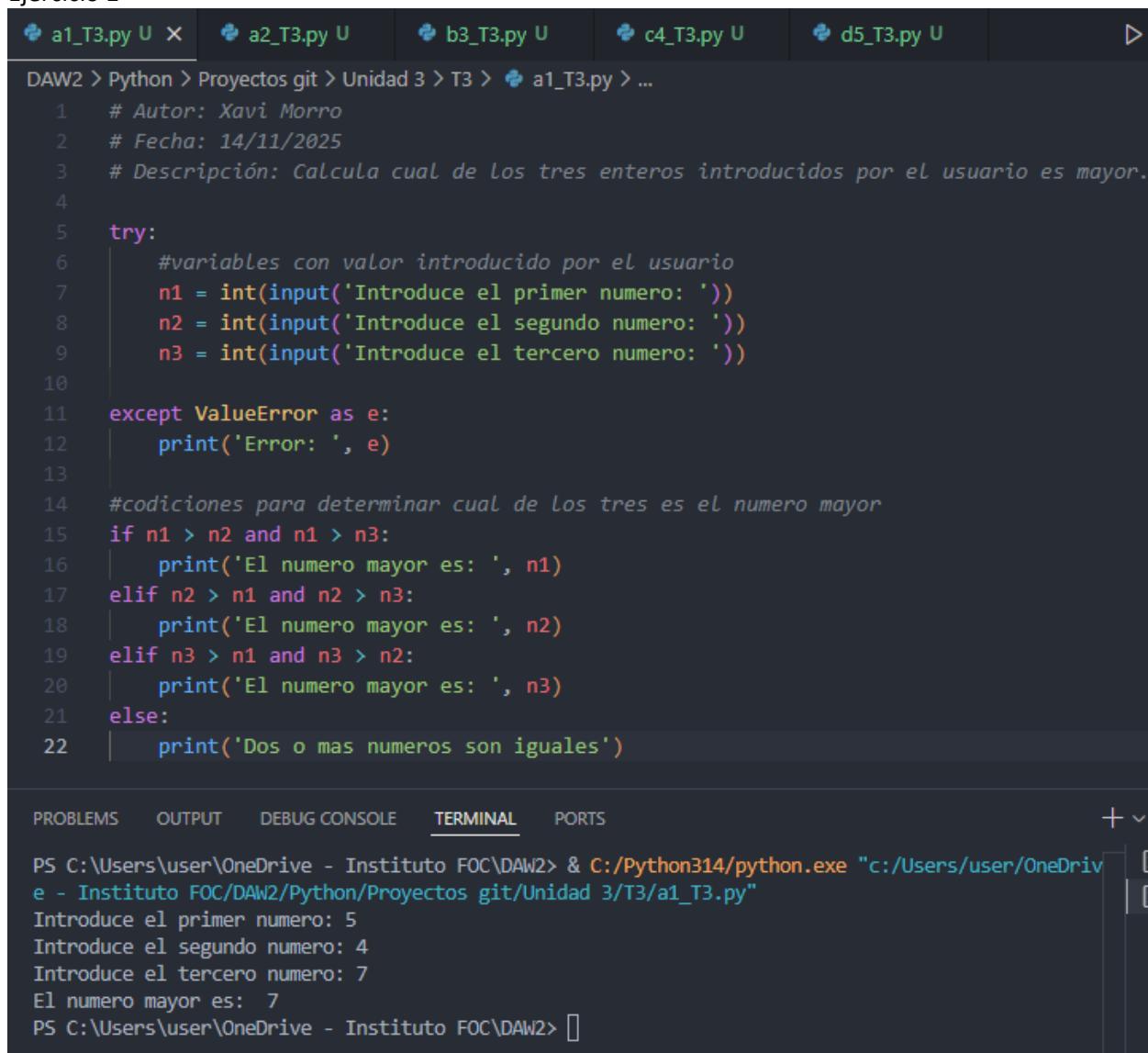
A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. ....

## 2. Resto de epígrafes que componen el desarrollo de este informe.

RA3\_a)

Ejercicio 1



```

a1_T3.py U X a2_T3.py U b3_T3.py U c4_T3.py U d5_T3.py U D
DAW2 > Python > Proyectos git > Unidad 3 > T3 > a1_T3.py > ...
1 # Autor: Xavi Morro
2 # Fecha: 14/11/2025
3 # Descripción: Calcula cual de los tres enteros introducidos por el usuario es mayor.
4
5 try:
6     #variables con valor introducido por el usuario
7     n1 = int(input('Introduce el primer numero: '))
8     n2 = int(input('Introduce el segundo numero: '))
9     n3 = int(input('Introduce el tercero numero: '))
10
11 except ValueError as e:
12     print('Error: ', e)
13
14 #condiciones para determinar cual de los tres es el numero mayor
15 if n1 > n2 and n1 > n3:
16     print('El numero mayor es: ', n1)
17 elif n2 > n1 and n2 > n3:
18     print('El numero mayor es: ', n2)
19 elif n3 > n1 and n3 > n2:
20     print('El numero mayor es: ', n3)
21 else:
22     print('Dos o mas numeros son iguales')

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS +

```

PS C:\Users\user\OneDrive - Instituto FOC\DAW2> & C:/Python314/python.exe "c:/Users/user/OneDrive - Instituto FOC/DAW2/Python/Proyectos git/Unidad 3/T3/a1_T3.py"
Introduce el primer numero: 5
Introduce el segundo numero: 4
Introduce el tercero numero: 7
El numero mayor es: 7
PS C:\Users\user\OneDrive - Instituto FOC\DAW2> []

```

## Ejercicio 2

The screenshot shows a code editor interface with several tabs at the top: a1\_T3.py U, a2\_T3.py U (selected), b3\_T3.py U, c4\_T3.py U, d5\_T3.py U, and a search/filter icon. The code in the editor is as follows:

```
DAW2 > Python > Proyectos git > Unidad 3 > T3 > a2_T3.py > ...
1 # Autor: Xavi Morro
2 # Fecha: 14/11/2025
3 # Descripción: Imprime los numeros enteros desde 1 hasta el numero elegido por el usuario.
4 try:
5     #se pide el valor de la variable al usuario
6     n1 = int(input('Introduce un numero entero positivo: '))
7
8 except ValueError as e:
9     print('Error: ', e);
10
11 #imprime todos Los numeros desde 1 hasta el introducido
12 if n1 > 0:
13     for i in range(1, n1 +1):
14         print(i)
```

Below the code editor is a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... | []
PS C:\Users\user\OneDrive - Instituto FOC\DAW2> & C:/Python314/python.exe "c:/Users/user/OneDrive - Instituto FOC/DAW2/Python/Proyectos git/Unidad 3/T3/a2_T3.py"
Introduce un numero entero positivo: 7
1
2
3
4
5
6
7
PS C:\Users\user\OneDrive - Instituto FOC\DAW2>
```

A dropdown menu on the right side of the terminal shows options: powershell and Python.

RA2\_b)

```

a1_T3.py U a2_T3.py U b3_T3.py U X c4_T3.py U d5_T3.py U
DAW2 > Python > Proyectos git > Unidad 3 > T3 > b3_T3.py > ...
1 # Autor: Xavi Morro
2 # Fecha: 14/11/2025
3 # Descripción: Calcula la operación elegida por el usuario de dos números
4 try:
5     #variables con valor introducido por el usuario
6     n1 = int(input('Introduce un numero: '))
7     n2 = int(input('Introduce un numero: '))
8     operacion = input('Introduce la operación a realizar: ')
9 except ValueError as e:
10     print('Error: ', e)
11
12 #posibles casos de la operación
13 if operacion == '+':
14     print(n1+n2)
15 elif operacion == '-':
16     print(n1-n2)
17 elif operacion == '*':
18     print(n1*n2)
19 elif operacion == '/':
20     print(n1/n2)
21 else:
22     print('Introduce una operación (+, -, *, /)')

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```

PS C:\Users\user\OneDrive - Instituto FOC\DAW2> & C:/Python314/python.exe "c:/Users/user/e - Instituto FOC/DAW2/Python/Proyectos git/Unidad 3/T3/b3_T3.py"
Introduce un numero: 5
Introduce un numero: 5
Introduce la operación a realizar: *
25
PS C:\Users\user\OneDrive - Instituto FOC\DAW2> []

```

## RA2\_c)

The screenshot shows the Visual Studio Code interface. At the top, there are tabs for files: a1\_T3.py, a2\_T3.py, b3\_T3.py, c4\_T3.py (which is the active tab), and d5\_T3.py. The main area displays the following Python code:

```

DAW2 > Python > Proyectos git > Unidad 3 > T3 > c4_T3.py > ...
1 # Autor: AVEL MORTO
2 # Fecha: 17/11/2025
3 # Descripción: Corrección del código para que calcule el área del rectángulo
4
5 def area_rectangulo(base, altura):
6     area = base * altura
7     return area
8
9 base = int(input('Introduce la base: '))
10 altura = int(input('Introduce la altura: '))
11 area = area_rectangulo(base, altura)
12
13 print('El área es: ', area)
14
15 #Error 1: Le falta una coma en el print.
16 #Error 2: Quitamos un * para que la operación se pueda hacer ya que el debugger da error.
17 #Error 3: Añadimos int a los inputs porque sino el debugger nos dice que no se pueden multiplicar dos strings.

```

The terminal below shows the execution of the script and its output:

```

PS C:\Users\user\OneDrive - Instituto FOC\DAW2> & C:/Python314/python.exe "c:/Users/user/OneDrive - Instituto FOC/DAW2/Python/Proyectos git/Unidad 3/T3/c4_T3.py"
Introduce la base: 5
Introduce la altura: 5
El área es: 25
PS C:\Users\user\OneDrive - Instituto FOC\DAW2>

```

This screenshot shows the VS Code interface with the debugger open. The terminal at the bottom shows the script being run and printing the area as 25.

The code editor shows the same Python script with several annotations:

- A red dot at line 7 indicates a breakpoint.
- Line 13 is highlighted with a yellow background, indicating it is the current line of execution.
- Annotations on the right side of the code editor highlight specific parts of the code:
  - "base = 5, altura = 5" is highlighted under line 11.
  - "area = 25" is highlighted under line 13.
  - "#Error 1: Le falta una coma en el print." is highlighted under line 15.
  - "#Error 2: Quitamos un \* para que la operación se pueda hacer ya que el debugger da error." is highlighted under line 16.
  - "#Error 3: Añadimos int a los inputs porque sino el debugger nos dice que no se" is highlighted under line 17.

This screenshot shows the terminal and the Python debugger interface. The terminal at the bottom shows the script being run and printing the area as 25.

The Python debugger interface on the right shows the following details:

- Session: powershell
- Session: Python
- Session: Python Deb...
- Session: Python Deb...

## RA2\_d)

```

DAW2 > Python > Proyectos git > Unidad 3 > T3 > d5_T3.py > ...
1 # Autor: Xavi Morro
2 # Fecha: 14/11/2025
3 # Descripción: Calcular la eficiencia en tiempo de dos sumas usando bucle for y sum(range).
4 import time
5
6 #primera operacion: iniciamos variable y guardamos el tiempo de inicio
7 # cuando acaba el for guarda el tiempo de nuevo y restamos los dos.
8 suma = 0;
9 inicio = time.time()
10 for i in range(1, 1000001):
11     suma += i
12 print(suma)
13 fin = time.time()
14 print('Primera op - Tiempo: ', fin - inicio, 'segundos')
15
16 #segunda operacion: hacemos lo mismo que en la primera pero con un sum(range()) asi comparamos cual es mas rapido.
17 inicio = time.time()
18 resul = sum(range(1, 1000001))
19 print(resul)
20 fin=time.time()
21 print('Segunda op - Tiempo: ', fin - inicio, 'segundos')

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\user\OneDrive - Instituto FOC\DAW2> & C:/Python314/python.exe "c:/Users/user/OneDrive - Instituto FOC/DAW2/Python/Proyectos git/Unidad 3/T3/d5\_T3.py"
500000500000
Primera op - Tiempo: 0.10262942314147949 segundos
500000500000
Segunda op - Tiempo: 0.015425443649291992 segundos
PS C:\Users\user\OneDrive - Instituto FOC\DAW2> []

Se puede apreciar claramente que el sum range es casi diez veces más eficiente.

## RA2\_e)

En las capturas se puede apreciar el código comentado y explicado correctamente.