

## Práctica 2: Creación y ejecución de programas Python

Datos personales

Apellidos: Diego González

Nombre: Adriel

### **1 Instrucciones del ejercicio 1**

```
>>> ((3*4)-(1*2))/((4*(-2)-(1*2)))
```

```
-1.0
```

```
>>> (2-(2*(-1)))/4
```

```
1.0
```

### **2 Instrucciones y resultados del ejercicio 2**

```
>>> a1 = 1
```

```
>>> b1 = -2
```

```
>>> c1 = 3
```

```
>>> a2 = 3.5
```

```
>>> b2 = -4
```

```
>>> c2 = 8.5
```

```
>>> y = (c1*a2-a1*c2)/(a2*b1-a1*b2)
```

```
>>> x = (c2-b2*y)/a2
```

```
...
```

```
>>> a1 = 1
```

```
>>> b1 = -2
```

```
>>> c1 = 3
```

```
>>> a2 = 4
```

```
>>> b2 = -8
```

```
>>> c2 = 7
```

```
>>> y = (c1*a2-a1*c2)/(a2*b1-a1*b2)
```

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

## Programación, Curso 2023-2024

ZeroDivisionError: division by zero

<pon aquí los resultados obtenidos al resolver los sistemas de ecuaciones propuestos>

```
>>> print(x,y)
```

```
1.6666666666666667 -0.6666666666666666
```

(El otro sistema al no tener soluciones no tiene sentido dar unas soluciones.)

¿Qué explicación puedes dar a los resultados?

El primer sistema es un sistema compatible determinado, por lo tanto tiene solución y el segundo es un sistema indeterminado, y no tiene solución. Al calcular la solución de una incógnita nos sale una división entre 0 lo cual no tiene ninguna solución.

### 3 Respuestas a las preguntas del ejercicio 4

¿Qué conclusiones extraes sobre la facilidad de entender el código con funciones?

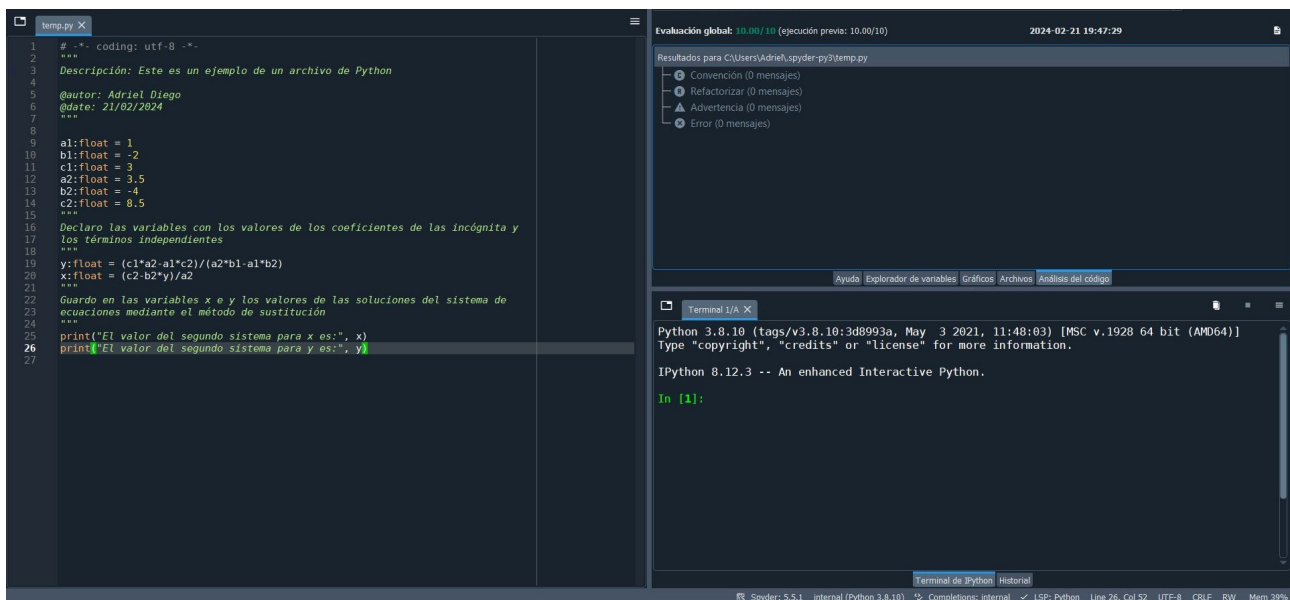
Permite optimizar los recursos y el tiempo a la hora de programar, ya que puedes de una forma intuitiva invocar partes de un script y poder cambiar los datos de entrada sin necesidad de rehacer el código.

¿Y sobre la repetición de código?

Que es una forma versátil de optimizar el tiempo y así dar el sentido de automatización de un programa, cambiando solo la información de las variables y repitiendo el código puedes obtener los resultados esperados sin gastar tiempo.

### 4 Parte avanzada: capturas de pantalla del análisis de código

<Si se han hecho, pon aquí las capturas de pantalla del análisis de código de los ejercicios 3, 4 y 5>



# Programación, Curso 2023-2024

The screenshot shows the Spyder IDE interface. The left pane displays a Python script named `temp.py` with the following content:

```
6 @autor: Adriel Diego
7 @date: 21/02/2024
8
9 """
10
11 def tejado():
12     """
13     Defino la función para imprimir el tejado""
14
15     """
16     print(" * * ")
17     print(" *** ")
18     print(" ***** ")
19     print(" * * * * * ")
20     print(" * * * * * ")
21     print(" * * * * * ")
22
23 def ventanas():
24     """
25     Defino la función para imprimir las ventanas
26
27     """
28     print("*****")
29     print("*** * * * *")
30     print("*** * * * *")
31     print("*****")
32
33 def puerta():
34     """
35     Defino la función para imprimir la puerta
36
37     """
38     print("*****")
39     print("*****")
40     print("*****")
41
42
43 tejado()
44 ventanas()
45 ventanas()
46 ventanas()
47 ventanas()
48 puerta()
49
```

The right pane shows the "Evaluación global" (Global Evaluation) window with a score of 10.00/10. Below it, the "Resultados para C:\Users\Adriel.spyder-py3\temp.py" window shows a list of messages: "Convención (0 mensajes)", "Refactorizar (0 mensajes)", "Advertencia (0 mensajes)", and "Error (0 mensajes)". The "Terminal de IPython" window shows the IPython prompt and the execution of the script, resulting in the output:

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.3 -- An enhanced Interactive Python.

In [1]:
```

The screenshot shows the Spyder IDE interface. The left pane displays a Python script named `temp.py` with the following content:

```
12 """
13 Defino la función para imprimir el tejado""
14
15 """
16 print(" * * ")
17 print(" *** ")
18 print(" ***** ")
19 print(" * * * * * ")
20 print(" * * * * * ")
21 print(" * * * * * ")
22
23 def ventanas():
24     """
25     Defino la función para imprimir las ventanas
26
27     """
28     print("*****")
29     print("*** * * * *")
30     print("*** * * * *")
31     print("*****")
32
33
34 def puerta():
35     """
36     Defino la función para imprimir la puerta
37
38     """
39     print("*****")
40     print("*****")
41     print("*****")
42
43 def main():
44     """
45     Defino la función principal
46
47     """
48     tejado()
49     ventanas()
50     ventanas()
51     ventanas()
52     puerta()
53
54 main()
55
```

The right pane shows the "Evaluación global" (Global Evaluation) window with a score of 10.00/10. Below it, the "Resultados para C:\Users\Adriel.spyder-py3\temp.py" window shows a list of messages: "Convención (0 mensajes)", "Refactorizar (0 mensajes)", "Advertencia (0 mensajes)", and "Error (0 mensajes)". The "Terminal de IPython" window shows the IPython prompt and the execution of the script, resulting in the output:

```
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.12.3 -- An enhanced Interactive Python.

In [1]:
```