

Prueba-Funcionalidad

Jose A.

Ejercicio 1

The screenshot shows a Python IDE with a project named 'ExamenSGE'. The file explorer on the left shows the project structure, including 'Ejercicios' and 'precio.txt'. The main editor displays the code for 'Ejercicio1.py'.

```
1 usage new *
2 def ConvertirBin(precio):
3     precio_final = ''
4
5     if precio:
6         precio_final = str(precio + 21)
7         precio = precio * 0.21
8     return str(precio) + precio_final
9
10 fich = open("precio.txt", "r")
11 dec = fich.read()
12 fich.close()
13
14 precio_final = ConvertirBin(int(dec))
15
16 fich = open("precio_final.txt", "w")
17 fich.write(precio_final)
18 fich.close()
```

Below the code editor, there are two terminal windows. The left terminal shows the output of the script, and the right terminal shows the output of the 'ConvertirBin()' function.

Precio inicial:

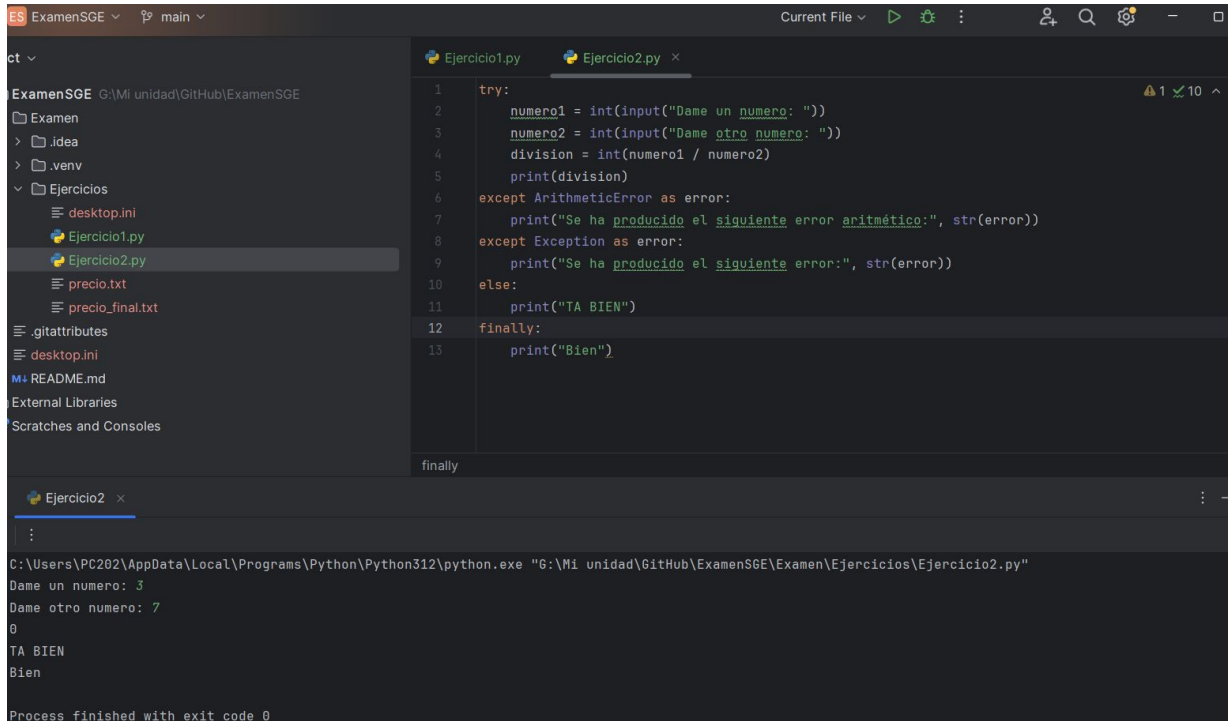
Archivo	Editar	Ver
100		

Precio final (el 121 es el resultado redondeado):

Archivo	Editar	Ver
21.0121		

The bottom status bar indicates the file path 'menSGE > Examen > Ejercicios > Ejercicio1.py', the time '7:38', the encoding 'CRLF', the file size 'UTF-8', the indentation '4 spaces', and the Python version 'Python 3.12'.

Ejercicio 2

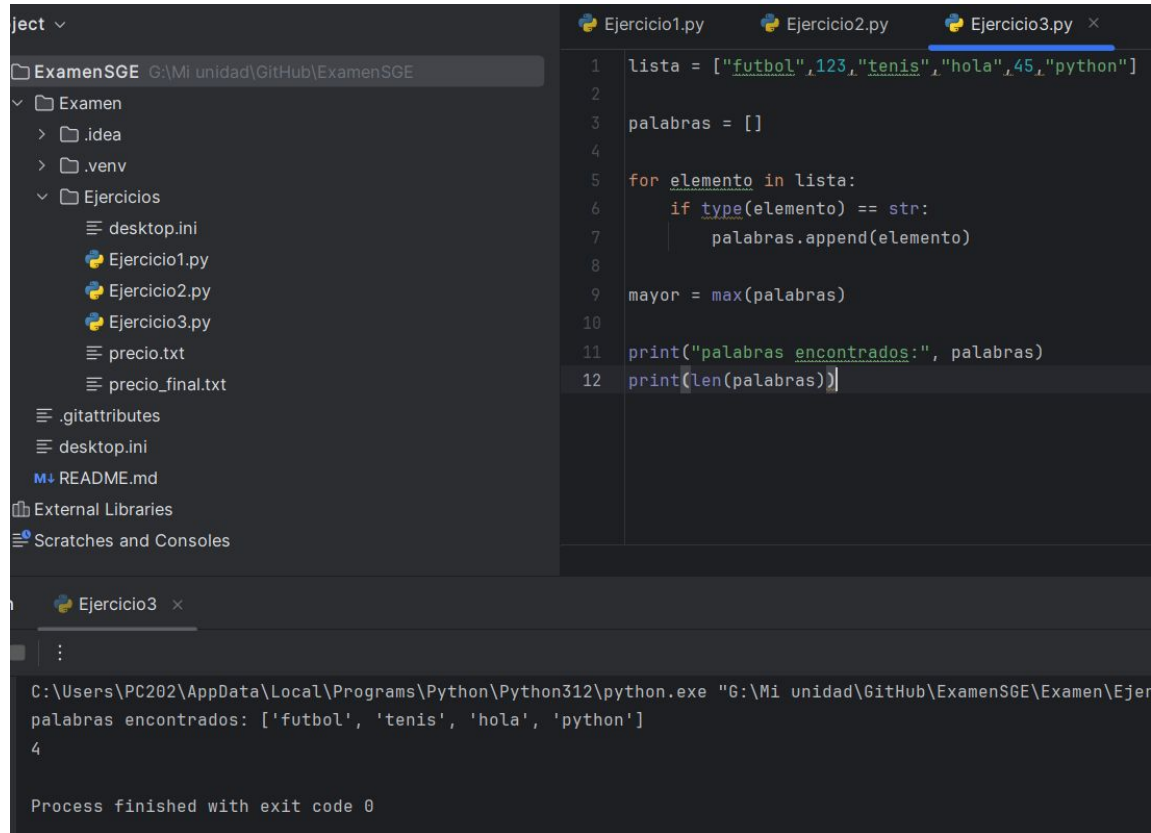


The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'ExamenSGE' with a subdirectory 'Ejercicios' containing 'Ejercicio1.py' and 'Ejercicio2.py'. The code editor displays the content of 'Ejercicio2.py', which is a Python script that prompts the user for two numbers and performs a division, handling potential arithmetic errors. The script uses a try-except-else-finally block. The output window at the bottom shows the execution of the script, displaying the prompts, the user input (3 and 7), the result (0), and the success message 'TA BIEN' and 'Bien'.

```
1 try:
2     numero1 = int(input("Dame un numero: "))
3     numero2 = int(input("Dame otro numero: "))
4     division = int(numero1 / numero2)
5     print(division)
6 except ArithmeticError as error:
7     print("Se ha producido el siguiente error aritmético:", str(error))
8 except Exception as error:
9     print("Se ha producido el siguiente error:", str(error))
10 else:
11     print("TA BIEN")
12 finally:
13     print("Bien")
```

Process finished with exit code 0

Ejercicio 3



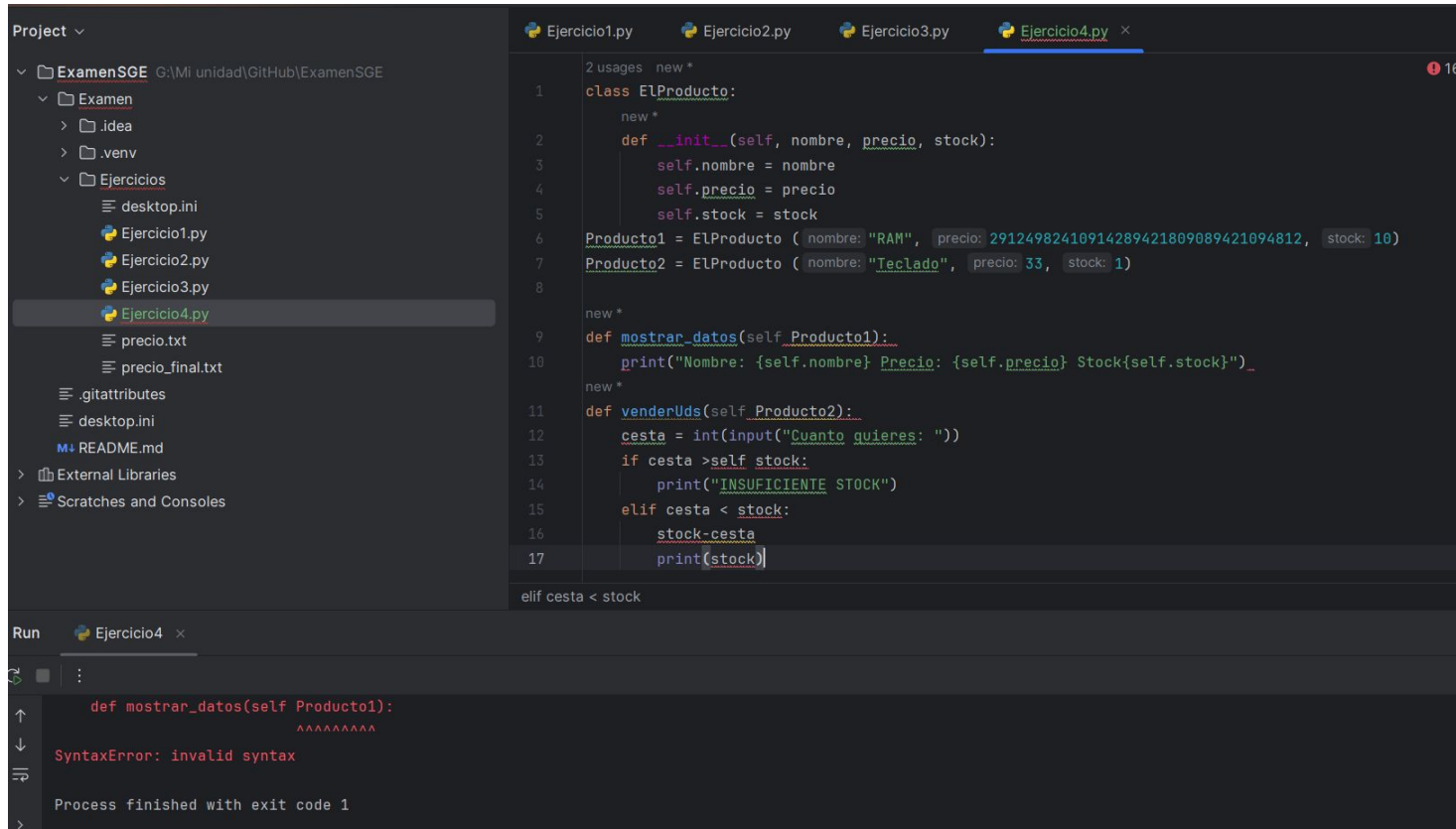
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'ExamenSGE' with a subdirectory 'Ejercicios' containing three Python files: 'Ejercicio1.py', 'Ejercicio2.py', and 'Ejercicio3.py'. The code editor displays the contents of 'Ejercicio3.py', which is a Python script that filters a list of strings from a mixed list. The script defines a list 'lista' containing strings and integers, initializes an empty list 'palabras', iterates through 'lista' to append only strings to 'palabras', and then prints the resulting list and its length. Below the code editor, a console window shows the output of the script: 'palabras encontrados: ['futbol', 'tenis', 'hola', 'python']' followed by the number '4' on a new line. The console also shows the command used to run the script and the message 'Process finished with exit code 0'.

```
1 lista = ["futbol", 123, "tenis", "hola", 45, "python"]
2
3 palabras = []
4
5 for elemento in lista:
6     if type(elemento) == str:
7         palabras.append(elemento)
8
9 mayor = max(palabras)
10
11 print("palabras encontrados:", palabras)
12 print(len(palabras))
```

C:\Users\PC202\AppData\Local\Programs\Python\Python312\python.exe "G:\Mi unidad\GitHub\ExamenSGE\Examen\Ejer
palabras encontrados: ['futbol', 'tenis', 'hola', 'python']
4

Process finished with exit code 0

Ejercicio 4 (No funciona)



The screenshot shows a Python IDE with a project named 'ExamenSGE'. The file explorer on the left shows the project structure, including 'Ejercicios' and 'Ejercicio4.py'. The main editor displays the code for 'Ejercicio4.py', which defines a class 'ElProducto' and two instances, 'Producto1' and 'Producto2'. The code also includes methods 'mostrar_datos' and 'venderUds'. The 'Run' panel at the bottom shows a 'SyntaxError: invalid syntax' error, indicating that the code is not syntactically correct.

```
2 usages new *
1 class ElProducto:
2     new *
3     def __init__(self, nombre, precio, stock):
4         self.nombre = nombre
5         self.precio = precio
6         self.stock = stock
7
8 Producto1 = ElProducto ( nombre: "RAM", precio: 29124982410914289421809089421094812, stock: 10)
9 Producto2 = ElProducto ( nombre: "Teclado", precio: 33, stock: 1)
10
11 new *
12 def mostrar_datos(self, Producto1):
13     print("Nombre: {self.nombre} Precio: {self.precio} Stock{self.stock}")
14
15 new *
16 def venderUds(self, Producto2):
17     cesta = int(input("Cuanto quieres: "))
18     if cesta > self.stock:
19         print("INSUFICIENTE STOCK")
20     elif cesta < stock:
21         stock-cesta
22         print(stock)
23
24 elif cesta < stock
```

Run Ejercicio4 x

```
def mostrar_datos(self, Producto1):
    print("Nombre: {self.nombre} Precio: {self.precio} Stock{self.stock}")
    print("Insuficiente Stock")
    print(stock-cesta)
    print(stock)
    print(cesta < stock)
```

SyntaxError: invalid syntax

Process finished with exit code 1