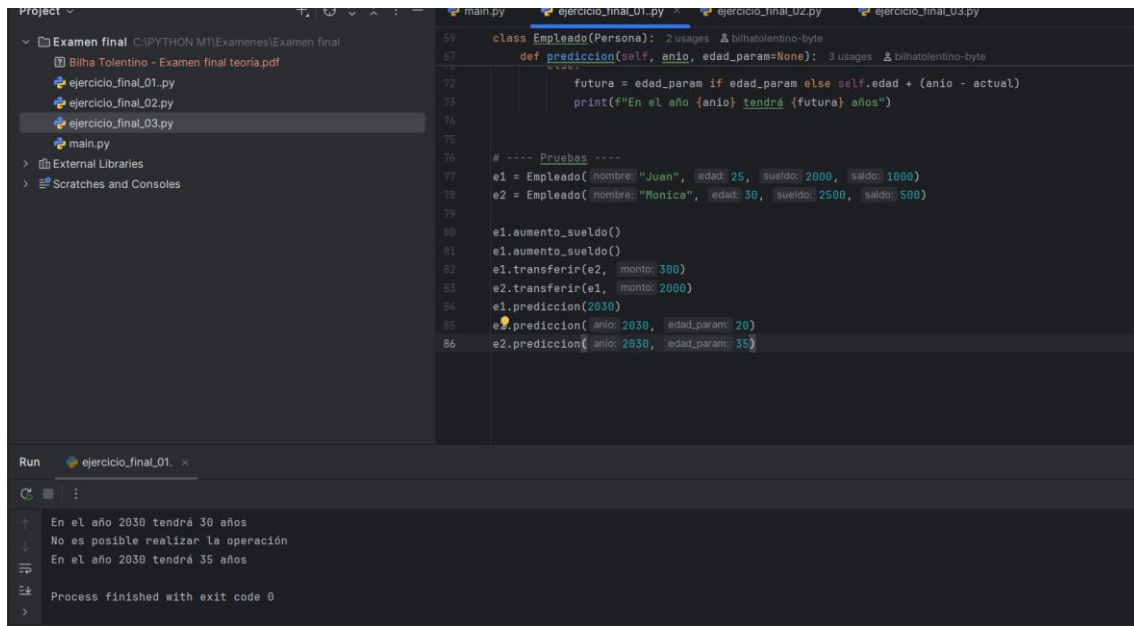


Ejercicio 1

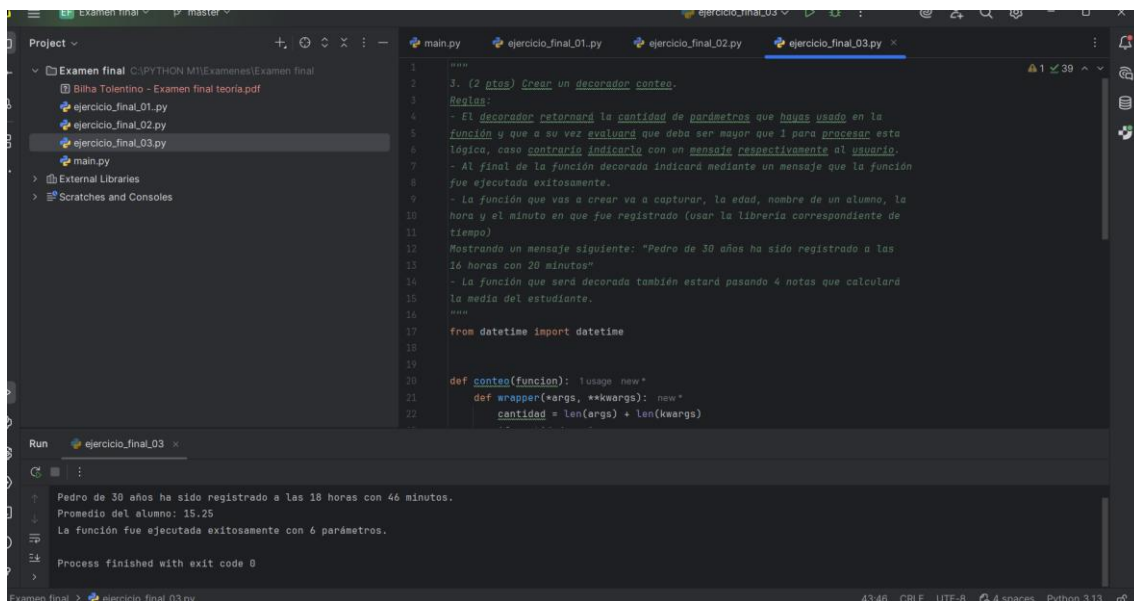


```
59 class Empleado(Persona):
60     def __init__(self, nombre, edad, sueldo, saldo):
61         self.nombre = nombre
62         self.edad = edad
63         self.sueldo = sueldo
64         self.saldo = saldo
65
66     def aumento_sueldo(self):
67         self.sueldo = self.sueldo * 1.1
68
69     def transferir(self, otro, monto):
70         if self.saldo < monto:
71             print("No es posible realizar la operación")
72         else:
73             self.saldo -= monto
74             otro.sueldo += monto
75
76     def prediccion(self, anio, edad_param=None):
77         futura = edad_param if edad_param else self.edad + (anio - actual)
78         print(f"En el año {anio} tendrá {futura} años")
79
80 # ---- Pruebas ----
81 e1 = Empleado(nombre="Juan", edad=25, sueldo=2000, saldo=1000)
82 e2 = Empleado(nombre="Monica", edad=30, sueldo=2500, saldo=500)
83
84 e1.aumento_sueldo()
85 e1.aumento_sueldo()
86 e1.transferir(e2, monto=300)
87 e2.transferir(e1, monto=2000)
88 e1.prediccion(2030)
89 e1.prediccion(anio=2030, edad_param=20)
90 e2.prediccion(anio=2030, edad_param=35)
```

Run ejercicio_final_01. x

```
En el año 2030 tendrá 30 años
No es posible realizar la operación
En el año 2030 tendrá 35 años
Process finished with exit code 0
```

Ejercicio 3



```
1 """
2 3. (2 pts) Crear un decorador contar.
3 Reglas:
4 - El decorador retornará la cantidad de parámetros que hayan usado en la
5 función y que a su vez evaluará que debe ser mayor que 1 para procesar esta
6 lógica, caso contrario indicará con un mensaje respectivamente al usuario.
7 - Al final de la función decorada indicará mediante un mensaje que la función
8 fue ejecutada exitosamente.
9 - La función que vas a crear va a capturar, la edad, nombre de un alumno, la
10 hora y el minuto en que fue registrado (usar la librería correspondiente de
11 tiempo)
12 Mostrando un mensaje siguiente: "Pedro de 30 años ha sido registrado a las
13 16 horas con 20 minutos"
14 - La función que será decorada también estará pasando 4 notas que calculará
15 la media del estudiante.
16 """
17 from datetime import datetime
18
19 def contar(funcion):
20     def wrapper(*args, **kwargs):
21         cantidad = len(args) + len(kwargs)
22         if cantidad < 1:
23             print("No es posible realizar la operación")
24         else:
25             funcion(*args, **kwargs)
26             print(f"La función fue ejecutada exitosamente con {cantidad} parámetros.")
27     return wrapper
```

Run ejercicio_final_03. x

```
Pedro de 30 años ha sido registrado a las 18 horas con 46 minutos.
Promedio del alumno: 15.25
La función fue ejecutada exitosamente con 6 parámetros.
Process finished with exit code 0
```