

Introducción a la programación en Python

Caso Práctico 2. Temas 6, 7, 8 y 9.

Autor: Francisco Mariño Ruiz

Alumno: Vicente Luis Cánovas Gil

Contenido

1	Pregunta 1. Tema 6 – Programación orientada a objetos.....	3
2	Pregunta 2. Tema 7 – Programación Funcional.....	4
3	Pregunta 3. Tema 8 – Excepciones.....	5
4	Pregunta 4. Tema 9 – Módulos y paquetes.....	6

1 Pregunta 1. Tema 6 – Módulos y paquetes

Escribe un script que cumpla las siguientes condiciones:

- Imprima por pantalla la hora con el siguiente formato: 19:37:52.
- Imprima por pantalla la ruta del ejecutable de Python.
- Imprima la versión de Python con la que se está trabajando.

Ejecuta el programa y copia también el resultado.

CÓDIGO PROPUESTO POR EL ALUMNO:

```
import datetime

import pathlib

import sys

esLaHora= datetime.datetime.now().time()

print("Hora actual: " + esLaHora.strftime('%H:%M:%S'))

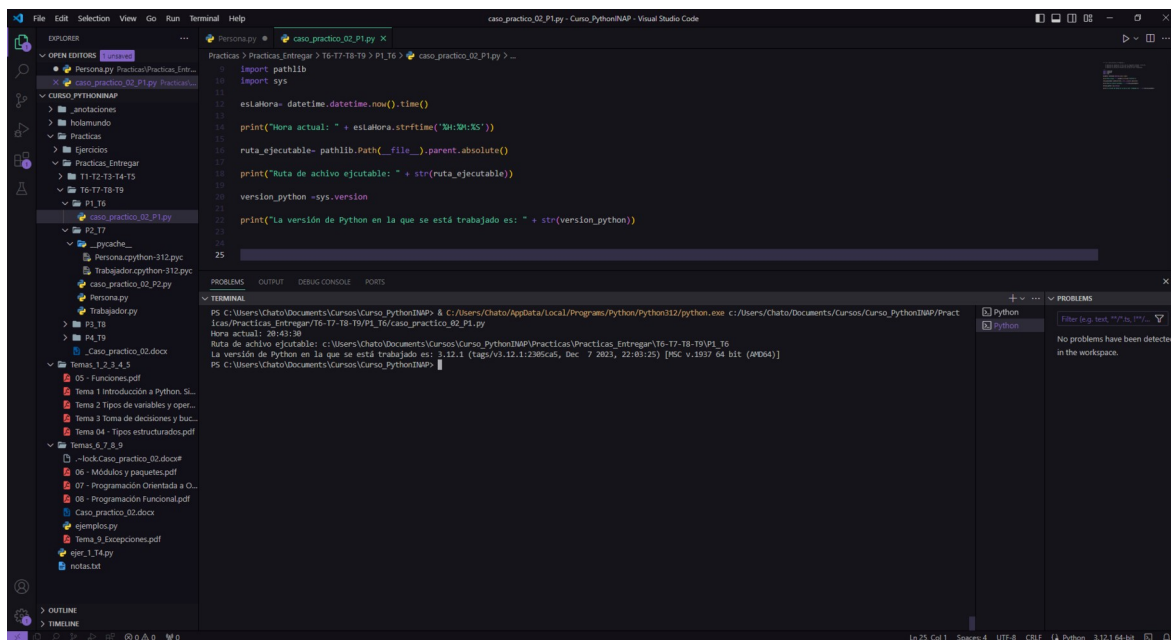
ruta_ejecutable= pathlib.Path(__file__).parent.absolute()

print("Ruta de achivo ejecutable: " + str(ruta_ejecutable))

version_python =sys.version

print("La versión de Python en la que se está trabajado es: " + str(version_python))
```

CAPTURA DE PANTALLA DE LA EJECUCIÓN DEL PROGRAMA:



2 Pregunta 2. Tema 7 – Programación orientada a objetos

Escribe un script que genere las siguientes clases nuevas:

- Clase “Persona”. Contendrá al menos los atributos DNI, nombre, apellidos. Tendrá un constructor y se personalizará el método “__str__” para ser impreso por pantalla.
- Clase “Trabajador” que herede de la clase “Persona” y añada al menos la propiedad “Cargo”. Tendrá un constructor y se personalizará el método “__str__”.

CÓDIGO PROPUESTO POR EL ALUMNO:

caso_practico_02_02.py

```
from Persona import Persona
```

```
from Trabajador import Trabajador
```

```
per = Persona ("Antonio Jose")
```

```
per.darApellido("Jimenez Martinez")
```

```
per.darDNI(23948345)
```

```
per.darFechaNacimiento ("12/04/1983")
```

```
trab = Trabajador("Oficinista")
```

```
trab.darSueldo(1200)
```

```
print("El trabajador " + str(per.mostrarNombre()) +
```

```
" " + str(per.mostrarApellidos())+ "con Dni " +str(per.mostrarDNI()) +
```

```
",\n con puesto en la empresa, " + str(trab.mostrarCargo()) +
```

```
",\n actualmente dispone de un salario de " + str( trab.mostrarSueldo()))
```

persona.py

```
class Persona():
```

```
    nombre=""  dni=""
```

```
    apellidos=""
```

```
    fechaNacimiento= ""
```

```
    def __init__(self, nombre):
```

```
        self.nombre= nombre
```

```
    def darNombre(self, nombre):
```

```
        self.nombre= nombre
```

```
    def darApellido(self, apellidos):
```

```
        self.apellidos= apellidos
```

```
    def darDNI(self, dni):
```

```
self.dni= dni

def darFechaNacimeitno(self, fechaNacimiento):

    self.fechaNacimiento= fechaNacimiento

def mostrarNombre(self):

    return self.nombre

def mostrarApellidos(self):

    return self.apellidos

def mostrarDNI(self):

    return self.dni

def mostrarFechaDeNacimiento(self):

    return self.fechaNacimiento
```

Trabajador.py

```
from Persona import Persona

class Trabajador():

    cargo=""

    sueldo=""

    def __init__(self,cargo):

        self.cargo = cargo

    def darCargo(self, cargo):

        self.cargo= cargo

    def darSueldo(self, sueldo):

        self.sueldo= sueldo

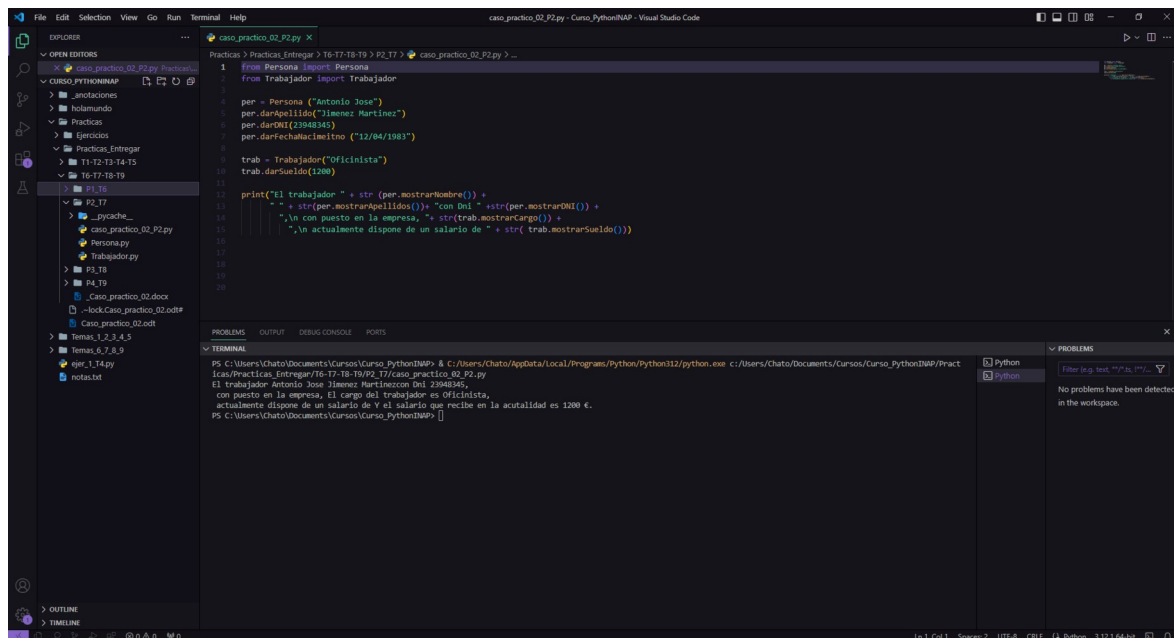
    def mostrarCargo(self):

        return "El cargo del trabajador es " + self.cargo

    def mostrarSueldo(self):

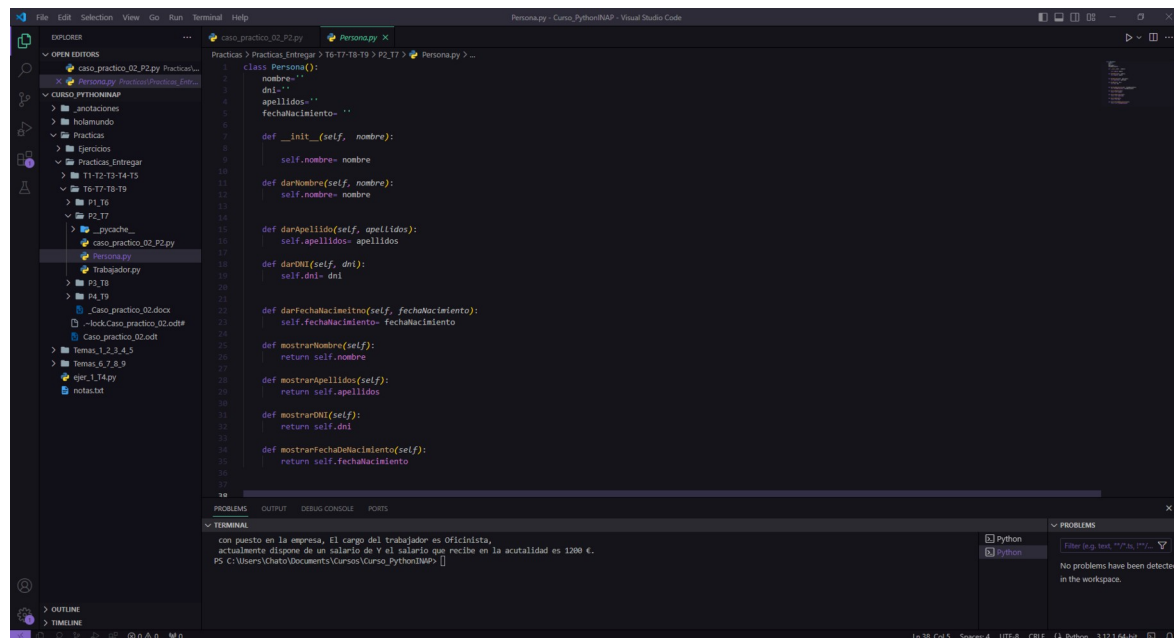
        return "Y el salario que recibe en la acutalidad es " + str(self.sueldo) + " €."
```

CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:



```
1 from Persona import Persona
2 from Trabajador import Trabajador
3
4 per = Persona("Antonio Jose")
5 per.darApellido("Jimenez Martinez")
6 per.darDNI(23948345)
7 per.darFechaNacimiento("12/04/1983")
8
9 trab = Trabajador("Oficinista")
10 trab.darSueldo(1200)
11
12 print("El trabajador " + str(per.mostrarNombre()) +
13       " " + str(per.mostrarApellidos()) + " con Dni " + str(per.mostrarDNI()) +
14       "\n con puesto en la empresa, " + str(trab.mostrarCargo()) +
15       "\n actualmente dispone de un salario de " + str(trab.mostrarSueldo()))
```

PS C:\Users\Chato\Documents\Cursos\Curso_PythonINAP> & c:\Users\Chato\AppData\Local\Programs\Python\Python312\python.exe c:\Users\Chato\Documents\Cursos\Curso_PythonINAP\Practicas\Practica_Entrega\T6-T7-T8-T9\T6-T7\T7\caso practico_02_P2.py
El trabajador Antonio Jose Jimenez Martinez con Dni 23948345,
con puesto en la empresa, El cargo del trabajador es Oficinista,
actualmente dispone de un salario de 1200 €.
PS C:\Users\Chato\Documents\Cursos\Curso_PythonINAP>



```
1 class Persona():
2     nombre=""
3     dni=""
4     apellidos=""
5     fechaNacimiento=""
6
7     def __init__(self, nombre):
8         self.nombre= nombre
9
10    def darNombre(self, nombre):
11        self.nombre= nombre
12
13    def darApellido(self, apellidos):
14        self.apellidos= apellidos
15
16    def darDNI(self, dni):
17        self.dni= dni
18
19    def darFechaNacimiento(self, fechaNacimiento):
20        self.fechaNacimiento= fechaNacimiento
21
22    def mostrarNombre(self):
23        return self.nombre
24
25    def mostrarApellidos(self):
26        return self.apellidos
27
28    def mostrarDNI(self):
29        return self.dni
30
31    def mostrarFechaNacimiento(self):
32        return self.fechaNacimiento
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

con puesto en la empresa, El cargo del trabajador es Oficinista,
actualmente dispone de un salario de 1200 €.
PS C:\Users\Chato\Documents\Cursos\Curso_PythonINAP>

The image shows a Visual Studio Code editor window with the following components:

- EXPLORER:** Displays a file tree with folders like 'Practicas', 'Ejercicios', and 'Practicas_Entregar'. The file 'caso_practico_02_P2.py' is selected.
- EDITOR:** Contains the Python code for the 'Persona' class. The code defines attributes (nombre, dni, apellidos, fechaNacimiento) and methods (__init__, darNombre, darApellido, darDNI, darFechaNacimiento, mostrarNombre, mostrarApellido, mostrarDNI, mostrarFechaNacimiento). Line 29 is highlighted.
- TERMINAL:** Shows the output of a command, indicating that the employee is an 'Oficinista' with a salary of 1200 €.
- PROBLEMS:** A panel at the bottom right showing no detected problems.

```
1 class Persona():
2     nombre=""
3     dni=""
4     apellidos=""
5     fechaNacimiento=""
6
7     def __init__(self, nombre):
8         self.nombre= nombre
9
10    def darNombre(self, nombre):
11        self.nombre= nombre
12
13    def darApellido(self, apellidos):
14        self.apellidos= apellidos
15
16    def darDNI(self, dni):
17        self.dni= dni
18
19    def darFechaNacimiento(self, fechaNacimiento):
20        self.fechaNacimiento= fechaNacimiento
21
22    def mostrarNombre(self):
23        return self.nombre
24
25    def mostrarApellido(self):
26        return self.apellidos
27
28    def mostrarDNI(self):
29        return self.dni
30
31    def mostrarFechaNacimiento(self):
32        return self.fechaNacimiento
33
34
35
36
37
```

con puesto en la empresa, El cargo del trabajador es Oficinista,
actualmente dispone de un salario de Y el salario que recibe en la actualidad es 1200 €.
PS C:\Users\Chato\Documents\Cursos\Curso_PythonINAP> []

3 Pregunta 3. Tema 8 – Programación Funcional

Escribe un script que contenga una función que tome una cadena como entrada y devuelva como salida la misma cadena sin el carácter 'o'.

Posteriormente, dada la cadena:

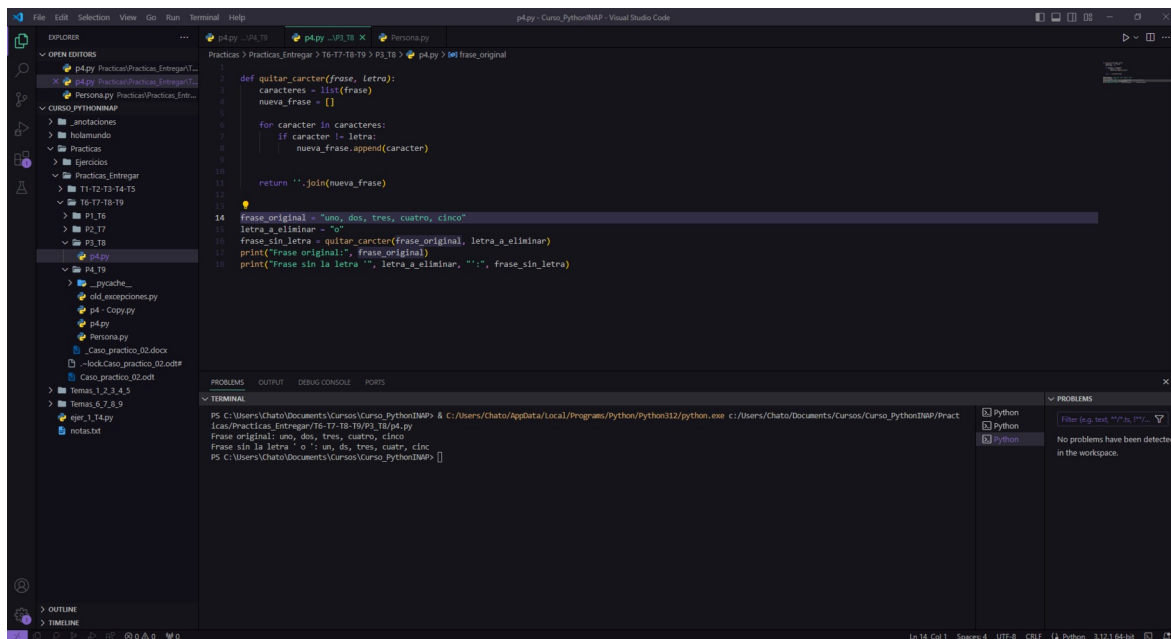
["uno", "dos", "tres", "cuatro", "cinco"]

aplica la función sobre dicha lista con un iterador de orden superior y copia el resultado.

CÓDIGO PROPUESTO POR EL ALUMNO:

```
def quitar_carcter(frase, letra):  
  
    caracteres = list(frase)  
  
    nueva_frase = []  
  
    for caracter in caracteres:  
  
        if caracter != letra:  
  
            nueva_frase.append(caracter)  
  
    return ''.join(nueva_frase)  
  
  
frase_original = "uno, dos, tres, cuatro, cinco"  
  
letra_a_eliminar = "o"  
  
frase_sin_letra = quitar_carcter(frase_original, letra_a_eliminar)  
  
print("Frase original:", frase_original)  
  
print("Frase sin la letra '", letra_a_eliminar, "':", frase_sin_letra)
```


CAPTURA DE PANTALLA DEL RESULTADO OBTENIDO:



The screenshot displays the Visual Studio Code interface with a Python file named `p4.py` open. The code defines a function `quitar_caracter` that removes a specified character from a string. The script then uses this function to process the string `"uno, dos, tres, cuatro, cinco"`, removing the character `"o"`. The output is printed to the terminal.

```
1 def quitar_caracter(frase, letra):
2     caracteres = list(frase)
3     nueva_frase = []
4
5     for caracter in caracteres:
6         if caracter != letra:
7             nueva_frase.append(caracter)
8
9     return ''.join(nueva_frase)
10
11
12
13
14 frase_original = "uno, dos, tres, cuatro, cinco"
15 letra_a_eliminar = "o"
16 frase_sin_letra = quitar_caracter(frase_original, letra_a_eliminar)
17 print("Frase original:", frase_original)
18 print("Frase sin la letra '", letra_a_eliminar, "'", frase_sin_letra)
```

The terminal output shows the execution results:

```
PS C:\Users\Chato\Documents\Cursos\Curso_Python\INAP> & C:\Users\Chato\AppData\Local\Programs\Python\Python312\python.exe c:\Users\Chato\Documents\Cursos\Curso_Python\INAP\Practicas\Practicas_Entregar\T6-T7-T8-T9\p3_T8\p4.py
Frase original: uno, dos, tres, cuatro, cinco
Frase sin la letra ' o ': un, ds, tres, cuatr, cinc
PS C:\Users\Chato\Documents\Cursos\Curso_Python\INAP>
```

4 Pregunta 4. Tema 9 – Excepciones

Dado el objeto “Persona” que se definió en la primera pregunta de este caso práctico, añadir a su constructor un control que en caso de que las propiedades “nombre” o “apellidos” no sean una cadena de caracteres, lance un error avisando de que el tipo de dato introducido es erróneo.

CÓDIGO PROPUESTO POR EL ALUMNO:

p4.py

```
from Persona import Persona
per = Persona ("Juan")
per.darNombre("chato")
per.darApellido= 4
per.darDNI= "23948345G"
per.darFechaNacimeitno ="12/04/1983"
print(per.mostrarNombre())
print(per.mostrarApellidos())
nom= str(per.mostrarNombre())
ape=per.mostrarApellidos()

def controdeTexto(self):

    if (self.isalpha()):

        return("la cadena es valida.")

    else:

        return("La cadena no reúne las condiciones exigidas. Tiene caracteres no validos.")

print("""+controdeTexto(nom)

print("""+controdeTexto(ape)
```

Personas.py

```
class Persona():

    nombre=""

    dni=""

    apellidos=""

    fechaNacimiento= ""

    texto=""

    def __init__(self, nombre):
```

```
self.nombre= nombre

def darNombre(self, nombre):

    self.nombre= nombre

def darApellido(self, apellidos):

    self.apellidos= apellidos

def darDNI(self, dni):

    self.dni= dni

def darFechaNacimeitno(self, fechaNacimiento):

    self.fechaNacimiento= fechaNacimiento


def mostrarNombre(self):

    return self.nombre


def mostrarApellidos(self):

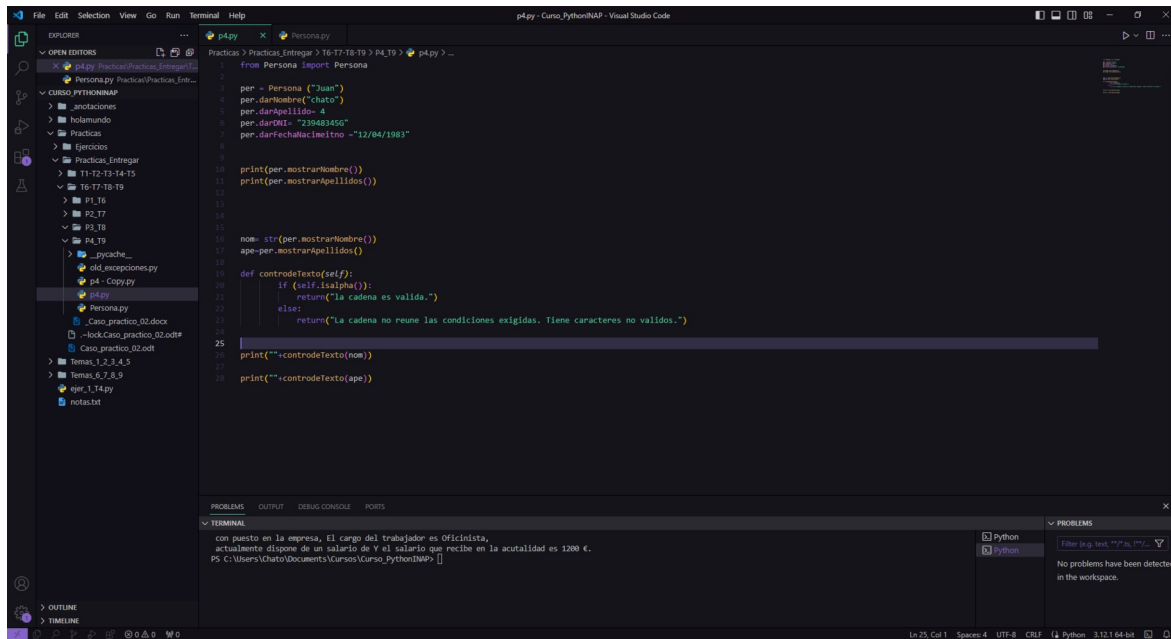
    return self.apellidos


def mostrarDNI(self):

    return self.dni


def mostrarFechaDeNacimiento(self):

    return self.fechaNacimiento
```

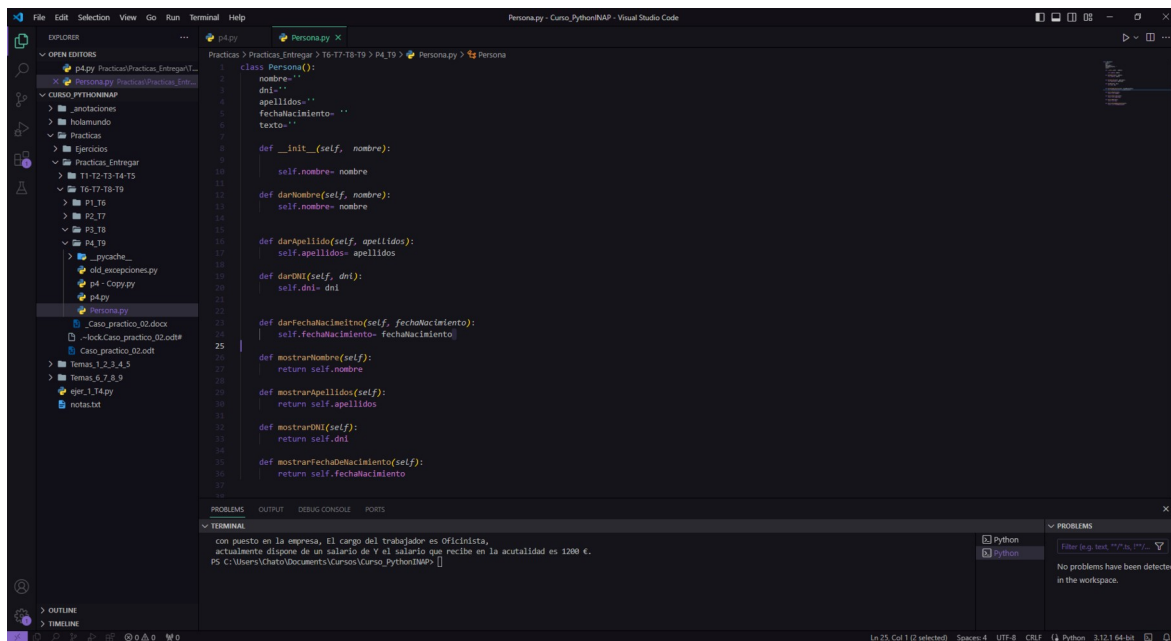


```
1 from Persona import Persona
2
3 per = Persona("Juan")
4 per.darNombre("chato")
5 per.darApellido= 4
6 per.darDni= "228823456"
7 per.darFechaNacimiento= "12/04/1983"
8
9
10 print(per.mostrarNombre())
11 print(per.mostrarApellidos())
12
13
14
15
16 nom= str(per.mostrarNombre())
17 ape=per.mostrarApellidos()
18
19 def contradeTexto(self):
20     if (self.isalpha()):
21         return("La cadena es valida.")
22     else:
23         return("La cadena no reúne las condiciones exigidas. Tiene caracteres no validos.")
24
25 print(contradeTexto(nom))
26
27 print(contradeTexto(ape))
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

TERMINAL

con puesto en la empresa, El cargo del trabajador es Oficinista,
actualmente dispone de un salario de Y el salario que recibe en la actualidad es 1200 €.
PS C:\Users\Chato\Documents\Cursos\Curso_Python\INAP> []



```
1 class Persona():
2     nombre=""
3     dni=""
4     apellidos=""
5     fechaNacimiento= ""
6     texto=""
7
8     def __init__(self, nombre):
9         self.nombre= nombre
10
11     def darNombre(self, nombre):
12         self.nombre= nombre
13
14
15     def darApellido(self, apellidos):
16         self.apellidos= apellidos
17
18     def darDni(self, dni):
19         self.dni= dni
20
21     def darFechaNacimiento(self, fechaNacimiento):
22         self.fechaNacimiento= fechaNacimiento
23
24     def mostrarNombre(self):
25         return self.nombre
26
27     def mostrarApellidos(self):
28         return self.apellidos
29
30     def mostrarDni(self):
31         return self.dni
32
33     def mostrarFechaNacimiento(self):
34         return self.fechaNacimiento
35
36
37
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

TERMINAL

con puesto en la empresa, El cargo del trabajador es Oficinista,
actualmente dispone de un salario de Y el salario que recibe en la actualidad es 1200 €.
PS C:\Users\Chato\Documents\Cursos\Curso_Python\INAP> []