

Nombre: Melanie Gusqui Pallo / Freddy Crofford Villao

Fecha: 26/11/2023

PROYECTO DE PROGRAMACION ORIENTADA A OBJETOS

The screenshot shows the Visual Studio Code interface with the 'ProgramacionObjeto' workspace selected. The current file is 'MATERIAL.py'. The code defines a class 'Material' that inherits from 'ABC'. It includes properties for 'codigo', 'autor', 'titulo', 'anio', 'editorial', and 'disponible', each with its own getter and setter methods. The code also contains comments for integrantes and a copyright notice.

```
from abc import ABC, abstractmethod

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

class Material(ABC):
    def __init__(self, codigo: str = None, autor: str = None, titulo: str = None, anio: int = None, editorial: str = None, disponible: bool = None, cantidad_disponible: int = None):
        self._codigo = codigo
        self._autor = autor
        self._titulo = titulo
        self._anio = anio
        self._editorial = editorial
        self._disponible = disponible
        self._cantidad_disponible = cantidad_disponible

    @property
    def codigo(self):
        return self._codigo

    @codigo.setter
    def codigo(self, nuevo_codigo):
        self._codigo = nuevo_codigo

    @property
    def autor(self):
        return self._autor

    @autor.setter
    def autor(self, nuevo_autor):
        self._autor = nuevo_autor

    @property
    def titulo(self):
        return self._titulo

    @titulo.setter
    def titulo(self, nuevo_titulo):
        self._titulo = nuevo_titulo

    @property
    def anio(self):
        return self._anio

    @anio.setter
    def anio(self, nuevo_anio):
        self._anio = nuevo_anio

Material > __init__()
```

This screenshot is identical to the one above, showing the 'MATERIAL.py' file in the Visual Studio Code interface. The code defines the 'Material' class with its properties and their getters and setters. The interface includes status bars at the bottom showing system information like date and weather.

```
1 usage
@property
def autor(self):
    return self._autor

1 usage
@author.setter
def autor(self, nuevo_autor):
    self._autor = nuevo_autor

1 usage
@property
def titulo(self):
    return self._titulo

1 usage
@titulo.setter
def titulo(self, nuevo_titulo):
    self._titulo = nuevo_titulo

1 usage
@property
def anio(self):
    return self._anio

@anio.setter
def anio(self, nuevo_anio):
    self._anio = nuevo_anio

Material > __init__()
```

A screenshot of a Windows desktop environment. In the center is a Microsoft Visual Studio Code window titled "MATERIAL.py". The code editor displays Python code for a class named "MATERIAL". The code includes several properties and their setters:

```
1 usage
44     @anio.setter
45     def anio(self, nuevo_anio):
46         self._anio = nuevo_anio
47
48     1 usage
49     @property
50     def editorial(self):
51         return self._editorial
52
53     1 usage
54     @editorial.setter
55     def editorial(self, nuevo_editorial):
56         self._editorial = nuevo_editorial
57
58     1 usage
59     @disponible.setter
60     def disponible(self):
61         return self._disponible
62
63     1 usage
64     @disponible.setter
65     def disponible(self, nuevo_estado):
66         self._disponible = nuevo_estado
67
68     1 usage
69     @cantidad_disponible.setter
70     def cantidad_disponible(self):
71         return self._cantidad_disponible
72
73     1 usage
74     @abstractmethod
75     def actualizar_disponibilidad(self, disponible):
76         pass
```

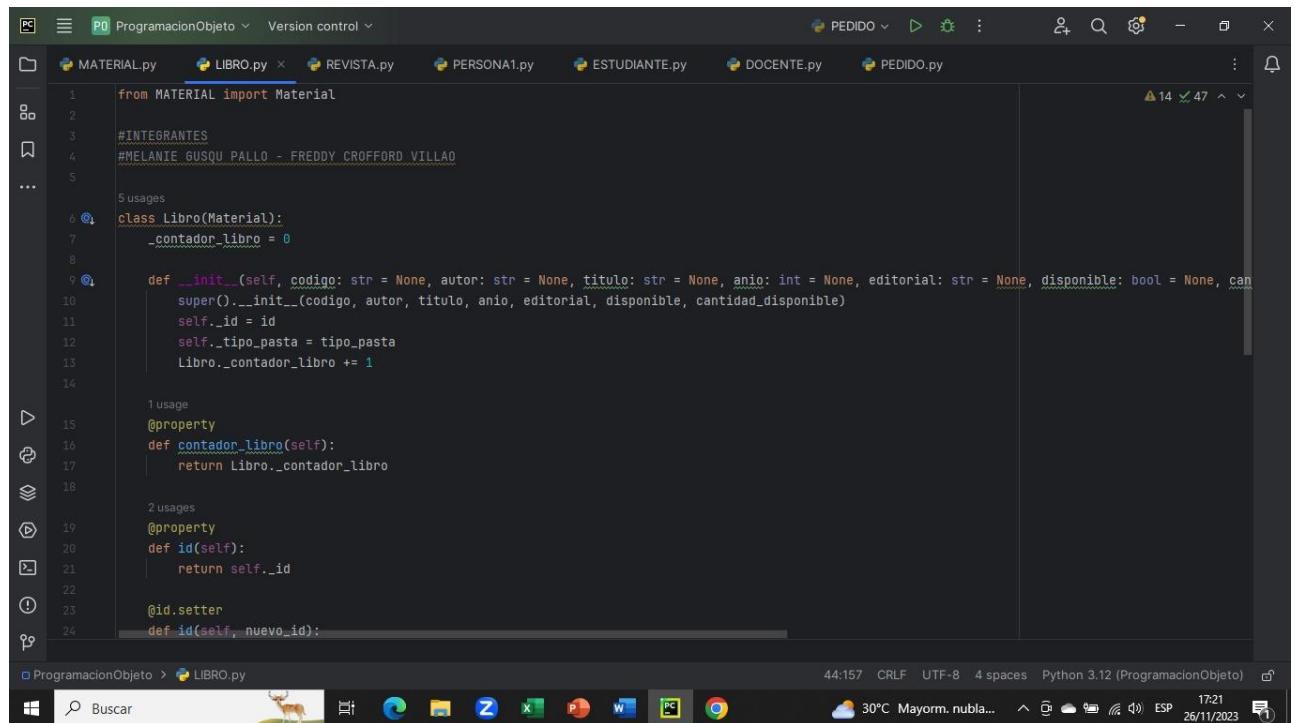
The status bar at the bottom of the code editor shows the file path "Material > __init__()", the file name "ProgramacionObjeto > MATERIAL.py", and the Python version "Python 3.12 (ProgramacionObjeto)". The system tray shows the date and time as "26/11/2023 17:20".

A screenshot of a Windows desktop environment, similar to the one above. In the center is a Microsoft Visual Studio Code window titled "MATERIAL.py". The code editor displays Python code for a class named "MATERIAL". The code includes several properties and their setters, with some changes compared to the first screenshot:

```
1 usage
60     @disponible.setter
61     def disponible(self, nuevo_estado):
62         self._disponible = nuevo_estado
63
64     1 usage
65     @property
66     def cantidad_disponible(self):
67         return self._cantidad_disponible
68
69     1 usage
70     @cantidad_disponible.setter
71     def cantidad_disponible(self, nueva_cantidad):
72         self._cantidad_disponible = nueva_cantidad
73
74     @abstractmethod
75     def actualizar_disponibilidad(self, disponible):
76         pass
```

The status bar at the bottom of the code editor shows the file path "Material > __init__()", the file name "ProgramacionObjeto > MATERIAL.py", and the Python version "Python 3.12 (ProgramacionObjeto)". The system tray shows the date and time as "26/11/2023 17:21".

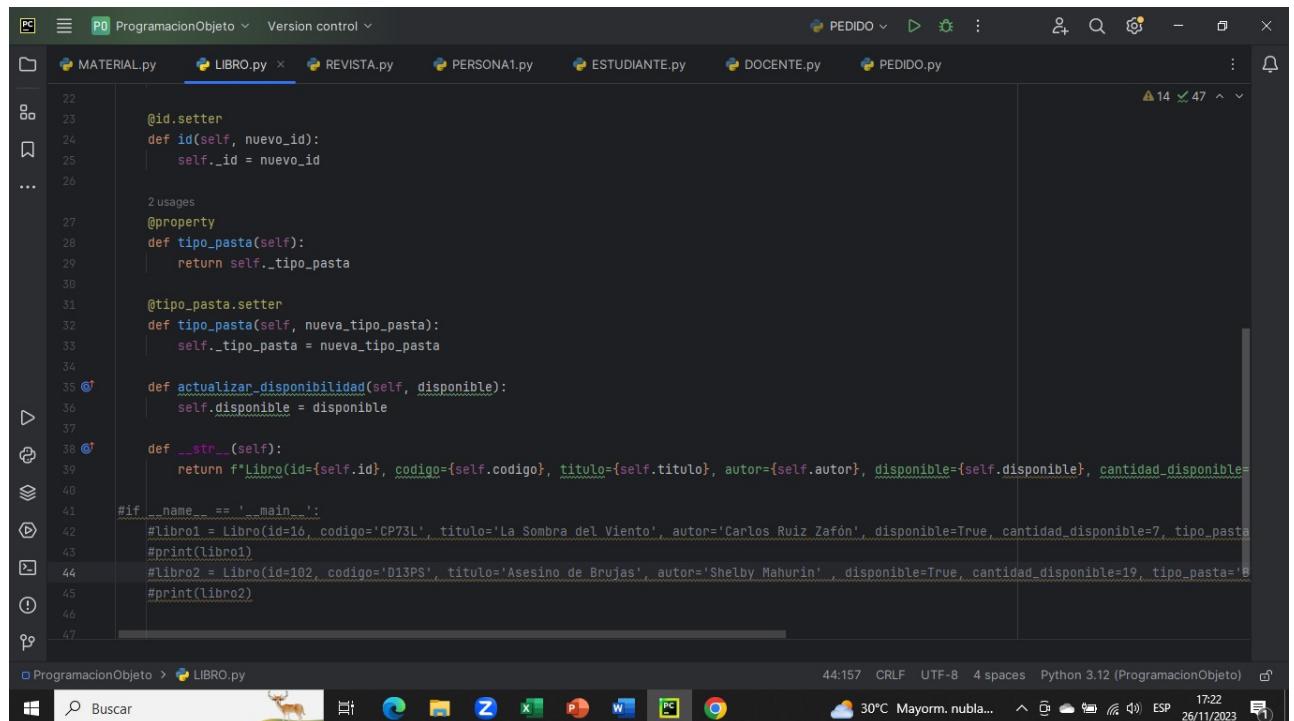
LIBRO



```
from MATERIAL import Material

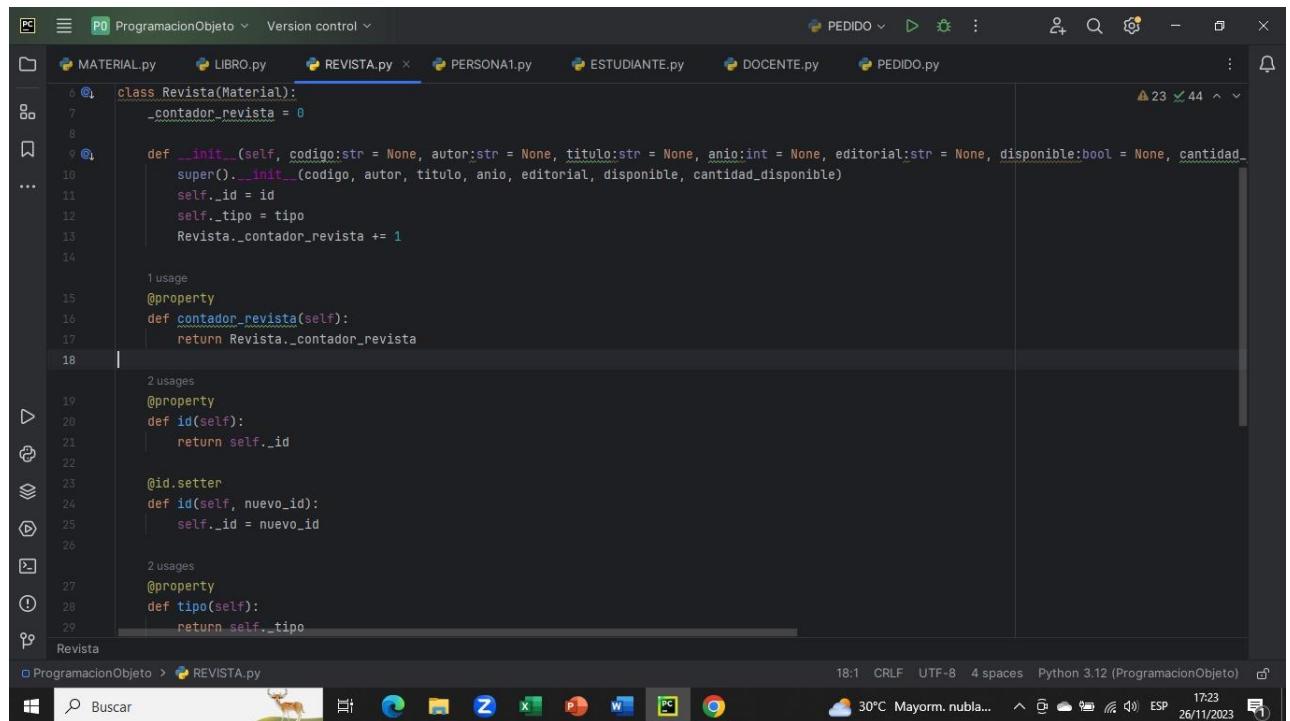
#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

5 usages
6 @
7     class Libro(Material):
8         _contador_libro = 0
9
10    def __init__(self, codigo: str = None, autor: str = None, titulo: str = None, anio: int = None, editorial: str = None, disponible: bool = None, cantidad_disponible: int = None):
11        super().__init__(codigo, autor, titulo, anio, editorial, disponible, cantidad_disponible)
12        self._id = id
13        self._tipo_pasta = tipo_pasta
14        Libro._contador_libro += 1
15
16    1 usage
17    @property
18    def contador_libro(self):
19        return Libro._contador_libro
20
21    2 usages
22    @property
23    def id(self):
24        return self._id
25
26    @id.setter
27    def id(self, nuevo_id):
28
29
30
31    @tipo_pasta.setter
32    def tipo_pasta(self, nueva_tipo_pasta):
33        self._tipo_pasta = nueva_tipo_pasta
34
35    def actualizar_disponibilidad(self, disponible):
36        self.disponible = disponible
37
38    def __str__(self):
39        return f"Libro(id={self.id}, codigo={self.codigo}, titulo={self.titulo}, autor={self.autor}, disponible={self.disponible}, cantidad_disponible={self.cantidad_disponible})"
40
41    #if __name__ == '__main__':
42    #libro1 = Libro(id=10, codigo='CP73L', titulo='La Sombra del Viento', autor='Carlos Ruiz Zafón', disponible=True, cantidad_disponible=7, tipo_pasta='B')
43    #print(libro1)
44    #libro2 = Libro(id=102, codigo='D13PS', titulo='Asesino de Brujas', autor='Shelby Mahurin', disponible=True, cantidad_disponible=19, tipo_pasta='B')
45    #print(libro2)
46
47
```



```
22
23    @id.setter
24    def id(self, nuevo_id):
25        self._id = nuevo_id
26
27    2 usages
28    @property
29    def tipo_pasta(self):
30        return self._tipo_pasta
31
32    @tipo_pasta.setter
33    def tipo_pasta(self, nueva_tipo_pasta):
34        self._tipo_pasta = nueva_tipo_pasta
35
36    def actualizar_disponibilidad(self, disponible):
37        self.disponible = disponible
38
39    def __str__(self):
40        return f"Libro(id={self.id}, codigo={self.codigo}, titulo={self.titulo}, autor={self.autor}, disponible={self.disponible}, cantidad_disponible={self.cantidad_disponible})"
41
42    #if __name__ == '__main__':
43    #libro1 = Libro(id=10, codigo='CP73L', titulo='La Sombra del Viento', autor='Carlos Ruiz Zafón', disponible=True, cantidad_disponible=7, tipo_pasta='B')
44    #print(libro1)
45    #libro2 = Libro(id=102, codigo='D13PS', titulo='Asesino de Brujas', autor='Shelby Mahurin', disponible=True, cantidad_disponible=19, tipo_pasta='B')
46
47
```

REVISTA



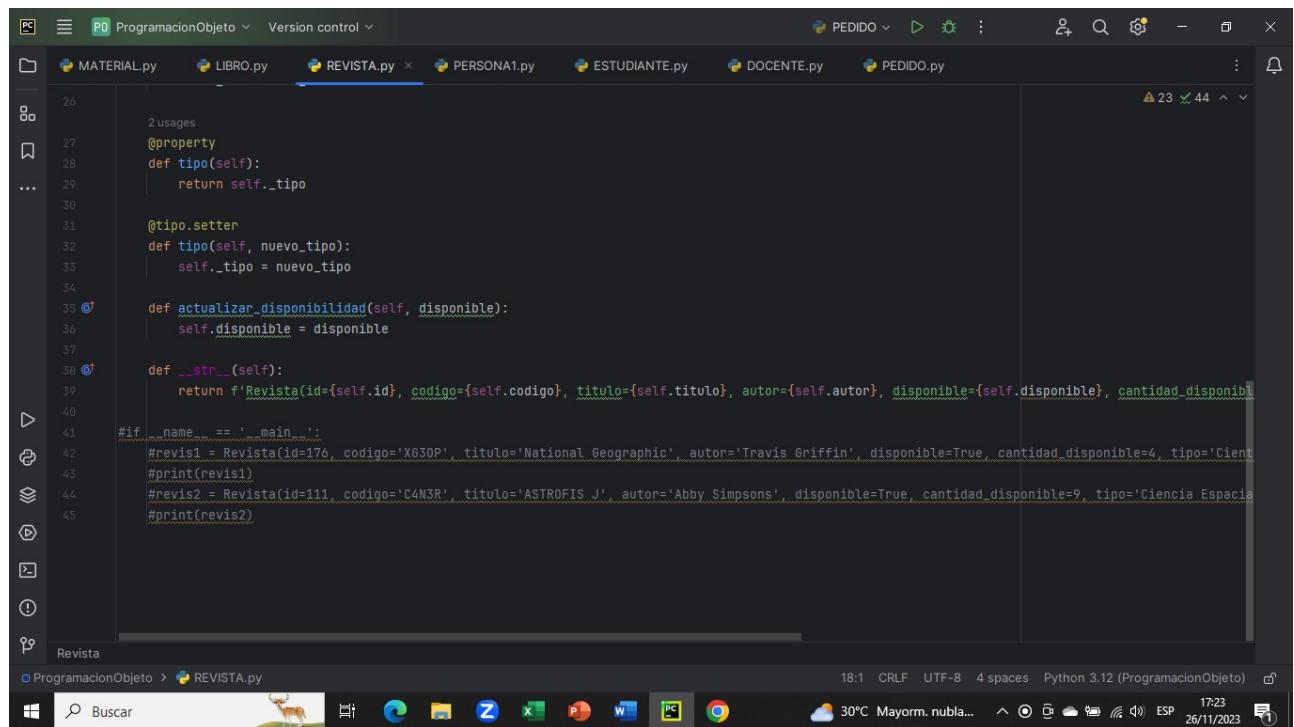
```
class Revista(Material):
    _contador_revista = 0

    def __init__(self, codigo:str = None, autor:str = None, titulo:str = None, anio:int = None, editorial:str = None, disponible:bool = None, cantidad_disponible:int = None):
        super().__init__(codigo, autor, titulo, anio, editorial, disponible, cantidad_disponible)
        self._id = id
        self._tipo = tipo
        Revista._contador_revista += 1

    @usage
    @property
    def contador_revista(self):
        return Revista._contador_revista

    @id.setter
    def id(self, nuevo_id):
        self._id = nuevo_id

    @tipo.setter
    def tipo(self):
        return self._tipo
```



```
2 usages
@property
def tipo(self):
    return self._tipo

@tipo.setter
def tipo(self, nuevo_tipo):
    self._tipo = nuevo_tipo

def actualizar_disponibilidad(self, disponible):
    self.disponible = disponible

def __str__(self):
    return f'Revista(id={self.id}, codigo={self.codigo}, titulo={self.titulo}, autor={self.autor}, disponible={self.disponible}, cantidad_disponible={self.cantidad_disponible})'

if __name__ == '__main__':
    #revis1 = Revista(id=170, codigo='XG3OP', titulo='National Geographic', autor='Travis Griffin', disponible=True, cantidad_disponible=4, tipo='Ciencia Espacial')
    #print(revis1)
    #revis2 = Revista(id=111, codigo='C4N3R', titulo='ASTROFIS J', autor='Abby Simpsons', disponible=True, cantidad_disponible=9, tipo='Ciencia Espacial')
    #print(revis2)
```

PERSONA

The screenshot shows the PyCharm IDE interface. The top navigation bar includes tabs for 'ProgramacionObjeto' and 'Version control'. The left sidebar contains icons for file operations like 'New', 'Open', 'Save', and 'Find'. The main code editor window displays Python code for a class 'PERSONA1.py'. The code defines a class 'Persona(ABC)' with an __init__ method and a __str__ method. It also includes a property 'cedula'. The status bar at the bottom shows the file path 'ProgramacionObjeto > PERSONA1.py', and system information like '22:1', 'CRLF', 'UTF-8', '4 spaces', 'Python 3.12 (ProgramacionObjeto)', '17:24', '30°C Mayorm. nubla...', and a date '26/11/2023'.

```
from abc import ABC, abstractmethod

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

4 usages
@ class Persona(ABC):
    def __init__(self, cedula :str, nombre :str, apellido :str, email :str, telefono :str, direccion :str,
                 numero_libros :int, activo :bool, carrera :str):
        self._cedula = cedula
        self._nombre = nombre
        self._apellido = apellido
        self._email = email
        self._telefono = telefono
        self._direccion = direccion
        self._numero_libros = numero_libros
        self._activo = activo
        self._carrera = carrera

    def __str__(self):
        return f'Persona: {str(self.__dict__)}'

3 usages
@property
def cedula(self):
    return self._cedula
```

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** P0 ProgramaciónObjeto > Version control
- File List:** MATERIAL.py, LIBRO.py, REVISTA.py, PERSONA1.py (highlighted), ESTUDIANTE.py, DOCENTE.py, PEDIDO.py
- Code Editor:** The PERSONA1.py file is open, showing Python code for a Person class with properties like cedula, nombre, apellido, and email.
- Code Navigator:** A sidebar on the left displays navigation links for the current file, including sections for cedula, nombre, apellido, and email.
- Status Bar:** Shows file statistics (54 usages, 49 dynamic), file size (17.24 MB), and system information (30°C, Mayorm. nubla...).

This screenshot shows a code editor window in Visual Studio Code (VS Code) with a dark theme. The current file is `PERSONA1.py`, which contains Python code for a `Persona` class. The code includes properties for `email`, `telefono`, `direccion`, and `numero_libros`, along with their respective getters and setters. A code navigation sidebar on the left lists these properties with their line numbers and usage counts. The status bar at the bottom shows the file path (`ProgramacionObjeto > PERSONA1.py`), Python version (`Python 3.12 (ProgramacionObjeto)`), and system information including temperature (`30°C`) and date (`26/11/2023`). The taskbar at the bottom also displays various application icons.

```
47     @email.setter
48     def email(self, email):
49         self._email = email
50
51     3 usages
52     @property
53     def telefono(self):
54         return self._telefono
55     2 usages
56     @telefono.setter
57     def telefono(self, telefono):
58         self._telefono = telefono
59
60     3 usages
61     @property
62     def direccion(self):
63         return self._direccion
64     2 usages
65     @direccion.setter
66     def direccion(self, direccion):
67         self._direccion = direccion
68
69     3 usages
70     @property
71     def numero_libros(self):
72         return self._numero_libros
73     2 usages
74
75     @numero_libros.setter
76     def numero_libros(self, numero_libros):
77         self._numero_libros = numero_libros
78
79     3 usages
80     @property
81     def activo(self):
82         return self._activo
83     2 usages
84     @activo.setter
85     def activo(self, activo):
86         self._activo = activo
87
88     3 usages
89     @property
90     def carrera(self):
91         return self._carrera
92     2 usages
93     @carrera.setter
94     def carrera(self, carrera):
95         self._carrera = carrera
96
97     @abstractmethod
98     def pedir_libro(self) -> bool:
99         pass
```

This screenshot shows the same code editor window in VS Code as the previous one, but with different code in `PERSONA1.py`. The new code defines properties for `numero_libros`, `activo`, `carrera`, and an abstract method `pedir_libro`. The code navigation sidebar on the left shows these properties and the abstract method with their line numbers and usage counts. The status bar and taskbar at the bottom remain the same, indicating the file path (`ProgramacionObjeto > PERSONA1.py`), Python version (`Python 3.12 (ProgramacionObjeto)`), and system information including temperature (`30°C`) and date (`26/11/2023`).

```
68     2 usages
69     @numero_libros.setter
70     def numero_libros(self, numero_libros):
71         self._numero_libros = numero_libros
72
73     3 usages
74     @property
75     def activo(self):
76         return self._activo
77     2 usages
78     @activo.setter
79     def activo(self, activo):
80         self._activo = activo
81
82     3 usages
83     @property
84     def carrera(self):
85         return self._carrera
86     2 usages
87     @carrera.setter
88     def carrera(self, carrera):
89         self._carrera = carrera
90
91     @abstractmethod
92     def pedir_libro(self) -> bool:
93         pass
```

PO ProgramacionObjeto Version control

MATERIAL.py LIBRO.py REVISTA.py PERSONA1.py ESTUDIANTE.py DOCENTE.py PEDIDO.py

54 49

```
77     self._activo = activo
78
79     3 usages
80     @property
81     def carrera(self):
82         return self._carrera
83
84     2 usages
85     @carrera.setter
86     def carrera(self, carrera):
87         self._carrera = carrera
88
89     @abstractmethod
90     def pedir_libro(self) -> bool:
91         pass
92
93     @abstractmethod
94     def devolver_libro(self) -> bool:
95         pass
96
97     if __name__ == '__main__':
98         pass
```

Person

ProgramacionObjeto > PERSONA1.py

22:1 CRLF UTF-8 4 spaces Python 3.12 (ProgramacionObjeto)

Buscar 30°C Mayorm. nubla... 17:25
ESP 26/11/2023

ESTUDIANTE

A screenshot of a Windows desktop with a Python code editor in Visual Studio Code. The file 'ESTUDIANTE.py' is open, showing the following code:

```
from PERSONA1 import Persona

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

class Estudiante(Persona):
    _contador_estudiante = 0

    def __init__(self, cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera, idES=int,
                 nivel=int):
        super().__init__(cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera)
        self.idES = idES
        self.nivel = nivel
        Estudiante._contador_estudiante += 1

    @property
    def idES(self):
        return self._idES

    @idES.setter
    def idES(self, idES):
        self._idES = idES

    @property
    def nivel(self):
        return self._nivel

    @nivel.setter
    def nivel(self, nivel):
        self._nivel = nivel

    def contadoc_estudiante(self):
        return Estudiante._contador_estudiante

    def pedir_libro(self):
        self._contador_estudiante += 1
        return True

    def devolver_libro(self):
        if self._contador_estudiante > 0:
            self._contador_estudiante -= 1
        return True
```

The status bar at the bottom indicates Python 3.12 (ProgramacionObjeto), along with system information like date and time.A screenshot of a Windows desktop with a Python code editor in Visual Studio Code. The file 'ESTUDIANTE.py' is open, showing the completed code from the previous screenshot:

```
from PERSONA1 import Persona

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

class Estudiante(Persona):
    _contador_estudiante = 0

    def __init__(self, cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera, idES=int,
                 nivel=int):
        super().__init__(cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera)
        self.idES = idES
        self.nivel = nivel
        Estudiante._contador_estudiante += 1

    @property
    def idES(self):
        return self._idES

    @idES.setter
    def idES(self, idES):
        self._idES = idES

    @property
    def nivel(self):
        return self._nivel

    @nivel.setter
    def nivel(self, nivel):
        self._nivel = nivel

    def contadoc_estudiante(self):
        return Estudiante._contador_estudiante

    def pedir_libro(self):
        self._contador_estudiante += 1
        return True

    def devolver_libro(self):
        if self._contador_estudiante > 0:
            self._contador_estudiante -= 1
        return True
```

The status bar at the bottom indicates Python 3.12 (ProgramacionObjeto), along with system information like date and time.

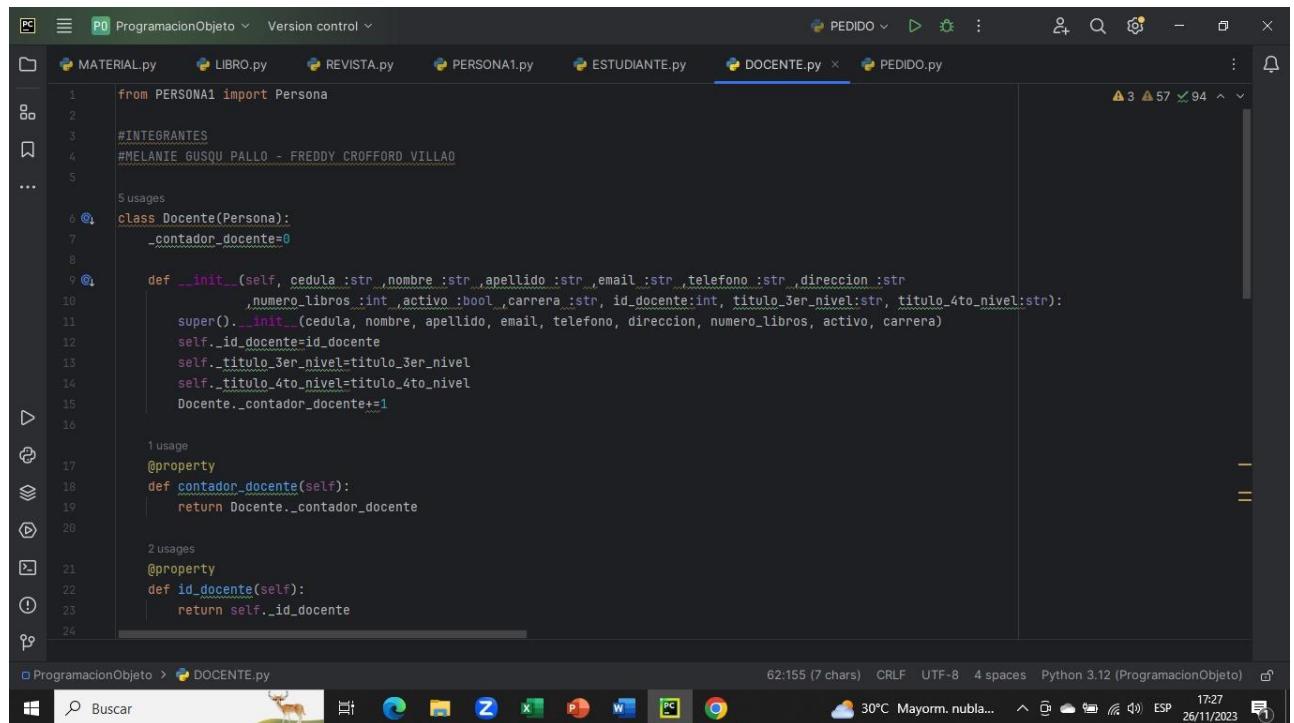
PC ProgramacionObjeto Version control

MATERIAL.py LIBRO.py REVISTA.py PERSONA1.py ESTUDIANTE.py DOCENTE.py PEDIDO.py

```
35
36     def pedir_libro(self):
37         self._contador_estudiante += 1
38         return True
39
40     def devolver_libro(self):
41         if self._contador_estudiante > 0:
42             self._contador_estudiante -= 1
43             return True
44         else:
45             return False
46
47     def __str__(self):
48         return f"Estudiante(cedula={self.cedula}, nombre={self.nombre}, apellido={self.apellido}, email={self.email}, telefono={self.telefono}, dirección={self.dirección})"
49
50 #if __name__ == "__main__":
51 #estudiante1 = Estudiante(cedula="0931054233", nombre="Melanie", apellido="Gusqui", email="melaniegusqui@gmail.com", telefono="0997079658", dirección="Av. 123, Ciudad Capital")
52 #print(estudiante1)
53 #estudiante1 = Estudiante(cedula="0957808181", nombre="Freddy", apellido="Crofford", email="crovillpapo@gmail.com", telefono="0978759219", dirección="Av. 456, Ciudad Nueva")
54 #print(estudiante1)
```

54:6 CRLF UTF-8 4 spaces Python 3.12 (ProgramacionObjeto) 17:26
Buscar 30°C Mayorm. nubla... 26/11/2023

DOCENTE



```
from PERSONA1 import Persona

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

class Docente(Persona):
    _contador_docente=0

    def __init__(self, cedula:str, nombre:str, apellido:str, email:str, telefono:str, direccion:str,
                 numero_libros:int, activo:bool, carrera:str, id_docente:int, titulo_3er_nivel:str, titulo_4to_nivel:str):
        super().__init__(cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera)
        self._id_docente=id_docente
        self._titulo_3er_nivel=titulo_3er_nivel
        self._titulo_4to_nivel=titulo_4to_nivel
        Docente._contador_docente+=1

    @property
    def contador_docente(self):
        return Docente._contador_docente

    @property
    def id_docente(self):
        return self._id_docente
```

```
def id_docente(self, nuevo_id):
    self._id_docente=nuevo_id

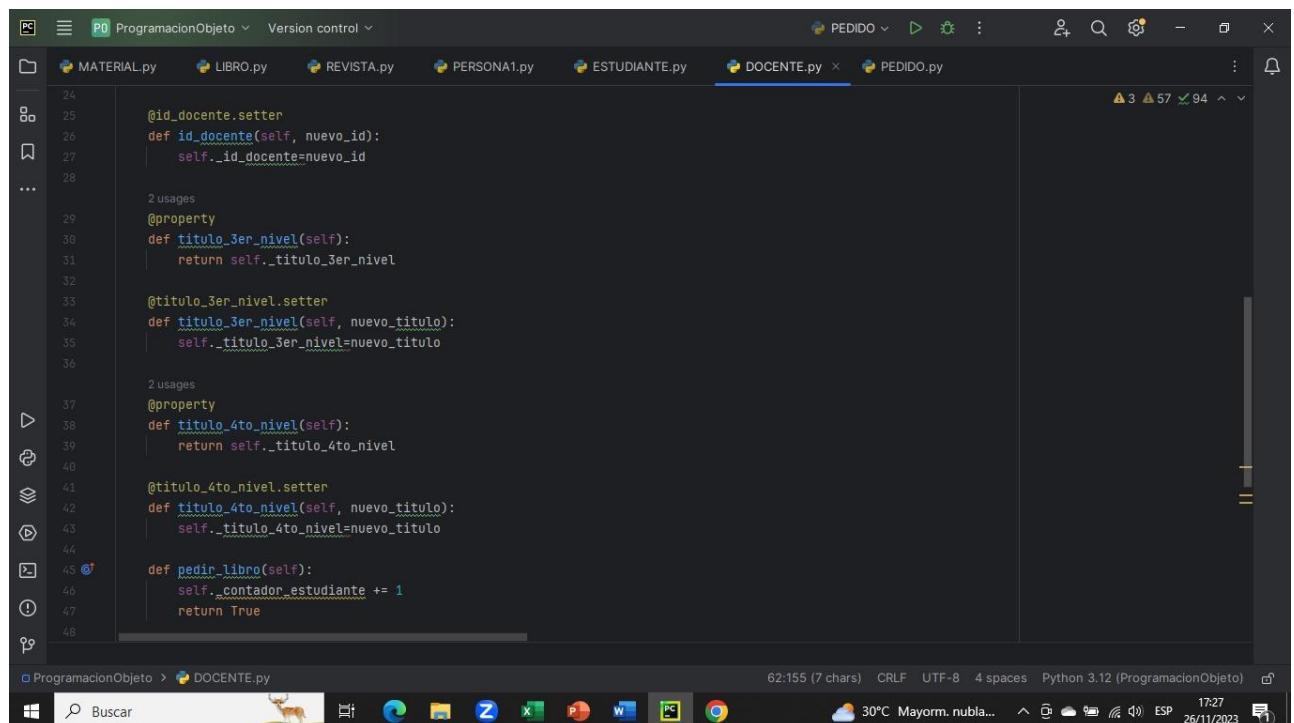
def titulo_3er_nivel(self):
    return self._titulo_3er_nivel

@titulo_3er_nivel.setter
def titulo_3er_nivel(self, nuevo_titulo):
    self._titulo_3er_nivel=nuevo_titulo

def titulo_4to_nivel(self):
    return self._titulo_4to_nivel

@titulo_4to_nivel.setter
def titulo_4to_nivel(self, nuevo_titulo):
    self._titulo_4to_nivel=nuevo_titulo

def pedir_libro(self):
    self._contador_estudiante += 1
    return True
```



```
def id_docente(self, nuevo_id):
    self._id_docente=nuevo_id

def titulo_3er_nivel(self):
    return self._titulo_3er_nivel

@titulo_3er_nivel.setter
def titulo_3er_nivel(self, nuevo_titulo):
    self._titulo_3er_nivel=nuevo_titulo

def titulo_4to_nivel(self):
    return self._titulo_4to_nivel

@titulo_4to_nivel.setter
def titulo_4to_nivel(self, nuevo_titulo):
    self._titulo_4to_nivel=nuevo_titulo

def pedir_libro(self):
    self._contador_estudiante += 1
    return True
```

The screenshot shows a Python code editor with the following code:

```
def pedir_libro(self):
    self._contador_estudiante += 1
    return True

def devolver_libro(self):
    if self._contador_estudiante > 0:
        self._contador_estudiante -= 1
    return True
    else:
        return False

def __str__(self):
    return f"Docente(cedula={self.cedula}, nombre={self.nombre}, apellido={self.apellido}, email={self.email}, telefono={self.telefono}, direccion={self.direccion})"

# if __name__ == "__main__":
#     docen1 = Docente(cedula="0690024187", nombre="Javier", apellido="Macias", email="javimacias@gmail.com", telefono="0995321105", direccion="Alborada")
#     print(docen1)
#     docen2 = Docente(cedula="0974321421", nombre="Mercedes", apellido="Mendoza", email="mercemend@hotmail.com", telefono="0987235117", direccion="Aca")
#     print(docen2)
```

PEDIDO

```
from LIBRO import Libro
from REVISTA import Revista
from ESTUDIANTE import Estudiante
from DOCENTE import Docente
from datetime import datetime

#INTEGRANTES
#MELANIE GUSQU PALLO - FREDDY CROFFORD VILLAO

3 usages
class Pedido(Libro, Revista, Estudiante, Docente):
    contador_pedido = 0

    def __init__(self, id_pedido=None, solicitante=None, lista_material=None, materia=None, fecha_prestamo=None, fecha_devolucion=None):
        self._id_pedido = id_pedido
        self._solicitante = solicitante
        self._lista_material = lista_material
        self._materia = materia
        self._fecha_prestamo = fecha_prestamo
        self._fecha_devolucion = fecha_devolucion
        self._nombre_solicitante = None
        Pedido.contador_pedido += 1

    @property
    def id_pedido(self):
        return self._id_pedido

if __name__ == '__main__':
    PEDIDO.py
```

```
@property
def solicitante(self):
    return self._solicitante

1 usage
@property
def lista_material(self):
    return self._lista_material

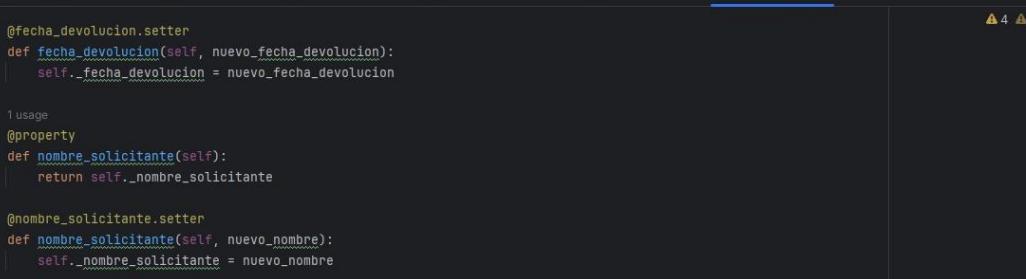
@lista_material.setter
def lista_material(self, nuevo_lista_material):
    self._lista_material = nuevo_lista_material

3 usages
@property
def fecha_prestamo(self):
    return self._fecha_prestamo

@fecha_prestamo.setter
def fecha_prestamo(self, nuevo_fecha_prestamo):
    self._fecha_prestamo = nuevo_fecha_prestamo

3 usages
@property
def fecha_devolucion(self):
    return self._fecha_devolucion

if __name__ == '__main__':
    PEDIDO.py
```



The screenshot shows a Windows desktop environment. In the foreground, there is a code editor window titled "PEDIDO.py" which contains Python code related to a library management system. The code includes methods for setting and getting the date of return, getting the borrower's name, and handling loan requests. It also includes methods for returning the material and printing a message about the loan. The code uses Python's object-oriented features like setters and getters. In the background, a web browser window is visible, showing a page with a deer logo and some text.

```
50 MATERIAL.py LIBRO.py REVISTA.py PERSONA1.py ESTUDIANTE.py DOCENTE.py PEDIDO.py
51
52     @fecha_devolucion.setter
53     def fecha_devolucion(self, nuevo_fecha_devolucion):
54         self._fecha_devolucion = nuevo_fecha_devolucion
55
56     @usage
57     @property
58     def nombre_solicitante(self):
59         return self._nombre_solicitante
60
61     @nombre_solicitante.setter
62     def nombre_solicitante(self, nuevo_nombre):
63         self._nombre_solicitante = nuevo_nombre
64
65     2 usages
66     def pedir_material(self, material):
67         self._lista_material = material
68         self._fecha_prestamo = datetime.now()
69         self._fecha_devolucion = None
70
71         if self._solicitante:
72             self._nombre_solicitante = self._solicitante.nombre
73             print(f"Se ha realizado el préstamo del material {self._lista_material} a {self._nombre_solicitante}.")
74
75     2 usages
76     def devolver_material(self):
77         if self._solicitante:
78             self._fecha_devolucion = datetime.now()
79
80
81 if __name__ == '__main__':
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
488
```

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** P0 ProgramacionObjeto > Version control
- Toolbars:** Standard Java-like icons for file operations.
- Code Editor:** The current file is `PEDIDO.py`. The code implements a `Pedido` class with methods for borrowing and returning materials. It also includes code snippets for `Revista`, `Libro`, and `Docente` classes.
- Status Bar:** Shows the time as 98:35, encoding as CRLF, character set as UTF-8, 4 spaces, Python 3.12 (ProgramacionObjeto), and the date as 26/11/2023.

```
73     self._fecha_devolucion = datetime.now()
74     print(f"Se ha devuelto el material {self._lista_material} de {self._nombre_solicitante}.")
75 else:
76     print("No hay material para devolver o no se ha registrado un solicitante.")
77
78
79 if __name__ == '__main__':
80     # Caso con Revista y Estudiante
81     revista1 = Revista(codigo=211, autor='Travis Griffin', titulo='National Geographic', anio=2021, editorial='Edelvives', disponible=True, cantidad_disponible=1)
82     estudiante = Estudiante(cedula='0931054233', nombre="Melanie", apellido="Gusqui", email="melaniegusqui@gmail.com", telefono="0997079658", direccion="Av. 123, Bogotá")
83
84     pedido1 = Pedido(solicitante=estudiante)
85
86     pedido1.pedir_material(revista1)
87     print(f"Fecha de préstamo: {pedido1.fecha_prestamo}")
88
89     pedido1.devolver_material()
90     print(f"Fecha de devolución: {pedido1.fecha_devolucion}")
91
92     # Caso con Libro y Docente
93     libro1 = Libro(codigo=101, autor='Shelby Mahurin', titulo='Asesino de Brujas', anio=2018, editorial='Puck', disponible=True, cantidad_disponible=19)
94     docente = Docente(cedula='0974321421', nombre="Mercedes", apellido="Mendoza", email="mercemend@hotmail.com", telefono="0987235117", direccion="Academia")
95
96     pedido2 = Pedido(solicitante=docente)
97
98     pedido2.pedir_material(libro1)
99     print(f"Fecha de préstamo: {pedido2.fecha_prestamo}")
```

If __name__ == '__main__':

The screenshot shows the PyCharm IDE interface. The code editor displays Python code for managing library items and requests. The run output window shows the execution results of the code. The system tray at the bottom right shows the date and time.

```
PO ProgramacionObjeto Version control
MATERIAL.py LIBRO.py REVISTA.py PERSONA1.py ESTUDIANTE.py DOCENTE.py PEDIDO.py
89     pedido1.devolver_material()
90     print(f"Fecha de devolución: {pedido1.fecha_devolucion}")
91
92     # Caso con Libro y Docente
93     libro1 = Libro(codigo=101, autor='Shelby Mahurin', titulo='Asesino de Brujas', anio=2018, editorial='Puck', disponible=True, cantidad_disponible=19)
94     docente = Docente(cedula='0974321421', nombre='Mercedes', apellido='Mendoza', email='mercemend@hotmail.com', telefono='09873235117', dirección='Acac
95
96     pedido2 = Pedido(solicitante=docente)
97
98     pedido2.pedir_material(libro1)
99     print(f"Fecha de préstamo: {pedido2.fecha_prestamo}")
if __name__ == '__main__':
    Run PEDIDO
    C:\Users\Usuario\PycharmProjects\ProgramacionObjeto\venv\Scripts\python.exe C:\Users\Usuario\PycharmProjects\ProgramacionObjeto\PEDIDO.py
    Se ha realizado el préstamo del material Revista(id=None, codigo=211, título=National Geographic, autor=Travis Griffin, disponible=True, cantidad_disponible=4, tipo=None)
    Fecha de préstamo: 2023-11-26 17:30:53.569382
    Se ha devuelto el material Revista(id=None, codigo=211, título=National Geographic, autor=Travis Griffin, disponible=True, cantidad_disponible=4, tipo=None)
    Fecha de devolución: 2023-11-26 17:30:53.569381
    Se ha realizado el préstamo del material Libro(id=None, codigo=101, título=Asesino de Brujas, autor=Shelby Mahurin, disponible=True, cantidad_disponible=19)
    Fecha de préstamo: 2023-11-26 17:30:53.569381
    Se ha devuelto el material Libro(id=None, codigo=101, título=Asesino de Brujas, autor=Shelby Mahurin, disponible=True, cantidad_disponible=19, tipo_pasta=None)
    Fecha de devolución: 2023-11-26 17:30:53.569381
Process finished with exit code 0
96:42 CRLF UTF-8 4 spaces Python 3.12 (ProgramacionObjeto) 17:31
30°C Mayorm. nubl... 26/11/2023
```