



Proyecto de programación

Integrantes:

Guaranda Baque Wendy.

Asunción Pin Kevin.

Fecha:

09/06/2024

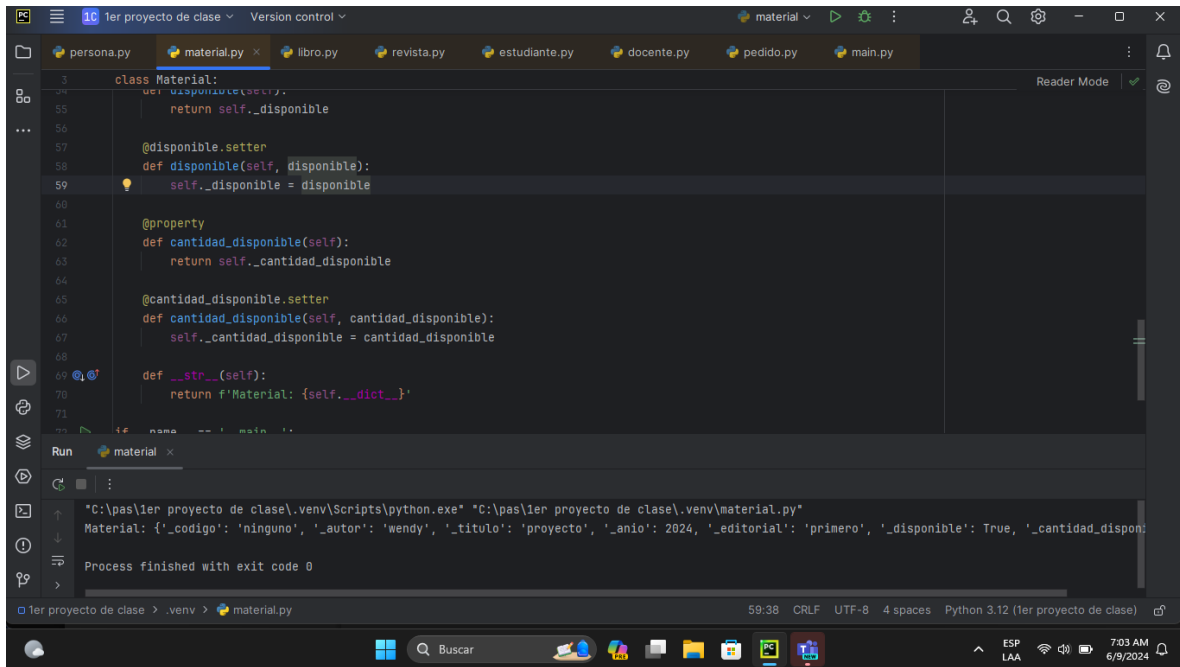
Class persona con su print.

```
1 class Persona:
2     def __init__(self, cedula=None, nombre=None, apellido=None, email=None, telefono=None,
3                 direccion=None, numero_libros=None, actividad=None, carrera=None):
4         self._cedula = cedula
5         self._nombre = nombre
6         self._apellido = apellido
7         self._email = email
8         self._telefono = telefono
9         self._direccion = direccion
10        self._numero_libros = numero_libros
11        self._actividad = actividad
12        self._carrera = carrera
13
14        @property
15        def cedula(self):
16            return self._cedula
17
18        @cedula.setter
19        def cedula(self, cedula):
```

Run persona x

```
"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\persona.py"
persona{'_cedula': '2987659503', '_nombre': 'Kevin', '_apellido': 'Asuncion', '_email': 'Kevin.paul@gmail.com', '_telefono': '593965748392', '_direccion': ''}
Process finished with exit code 0
```

Class material con su print.



The screenshot shows a VS Code editor with the file explorer on the left displaying a project named '1er proyecto de clase'. The file 'material.py' is open in the editor. The code defines a class 'Material' with attributes 'codigo', 'autor', 'titulo', 'anio', 'editorial', 'disponible', and 'cantidad_disponible'. It includes methods for setting and getting these attributes, and a '__str__' method for printing the object's state. Below the editor, the Run and Debug console shows the output of running 'material.py', displaying the object's state as a dictionary. The status bar at the bottom indicates the file is 'material.py' in the project directory, using Python 3.12.

```
class Material:
    def __init__(self, codigo=None, autor=None, titulo=None, anio=2024, editorial='primero', disponible=True, cantidad_disponible=0):
        self.codigo = codigo
        self.autor = autor
        self.titulo = titulo
        self.anio = anio
        self.editorial = editorial
        self.disponible = disponible
        self.cantidad_disponible = cantidad_disponible

    @disponible.setter
    def disponible(self, disponible):
        self._disponible = disponible

    @property
    def cantidad_disponible(self):
        return self._cantidad_disponible

    @cantidad_disponible.setter
    def cantidad_disponible(self, cantidad_disponible):
        self._cantidad_disponible = cantidad_disponible

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

if __name__ == '__main__':
    material = Material(codigo='ninguno', autor='wendy', titulo='proyecto', anio=2024, editorial='primero', disponible=True, cantidad_disponible=0)
    print(material)
```

Run material x

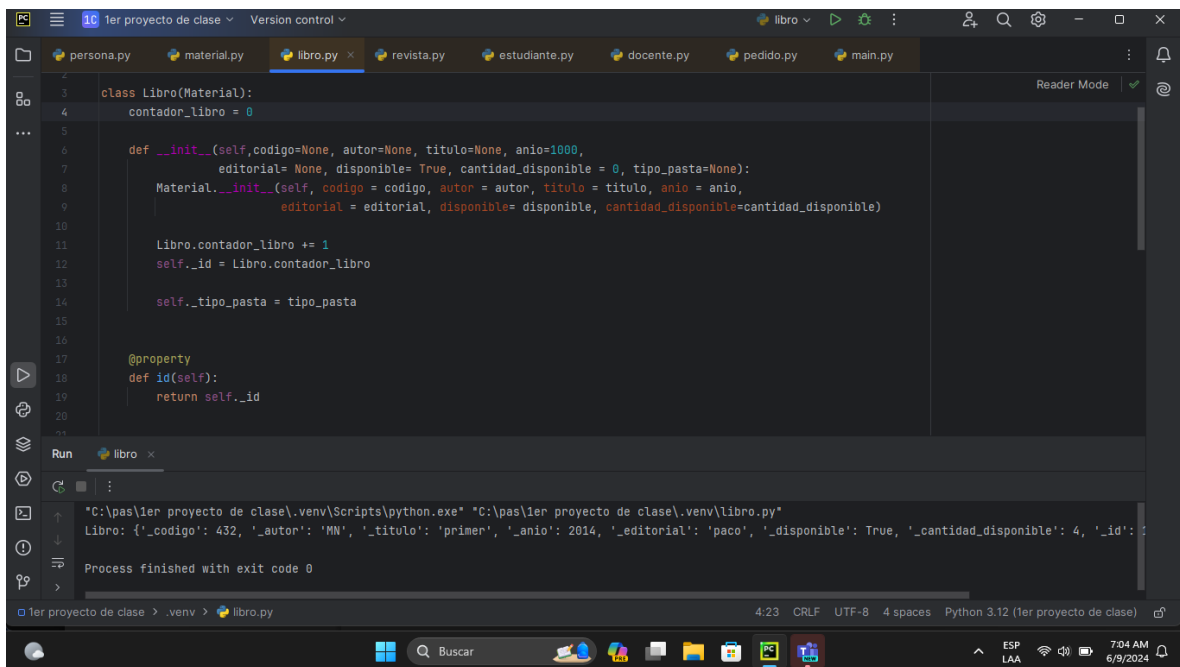
"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\material.py"

Material: {'_codigo': 'ninguno', '_autor': 'wendy', '_titulo': 'proyecto', '_anio': 2024, '_editorial': 'primero', '_disponible': True, '_cantidad_disponible': 0}

Process finished with exit code 0

1er proyecto de clase > .venv > material.py 59:38 CRLF UTF-8 4 spaces Python 3.12 (1er proyecto de clase)

Class libro con su print.



The screenshot shows a VS Code editor with the file explorer on the left displaying a project named '1er proyecto de clase'. The file 'libro.py' is open in the editor. The code defines a class 'Libro' that inherits from 'Material'. It has a class attribute 'contador_libro' and a method '__init__' that initializes the object's attributes and increments the counter. It also has a property 'id' that returns the object's unique identifier. Below the editor, the Run and Debug console shows the output of running 'libro.py', displaying the object's state as a dictionary. The status bar at the bottom indicates the file is 'libro.py' in the project directory, using Python 3.12.

```
class Libro(Material):
    contador_libro = 0

    def __init__(self, codigo=None, autor=None, titulo=None, anio=2014, editorial='paco', disponible=True, cantidad_disponible=0, tipo_pasta=None):
        Material.__init__(self, codigo=codigo, autor=autor, titulo=titulo, anio=anio, editorial=editorial, disponible=disponible, cantidad_disponible=cantidad_disponible)

        Libro.contador_libro += 1
        self.id = Libro.contador_libro
        self.tipo_pasta = tipo_pasta

    @property
    def id(self):
        return self._id
```

Run libro x

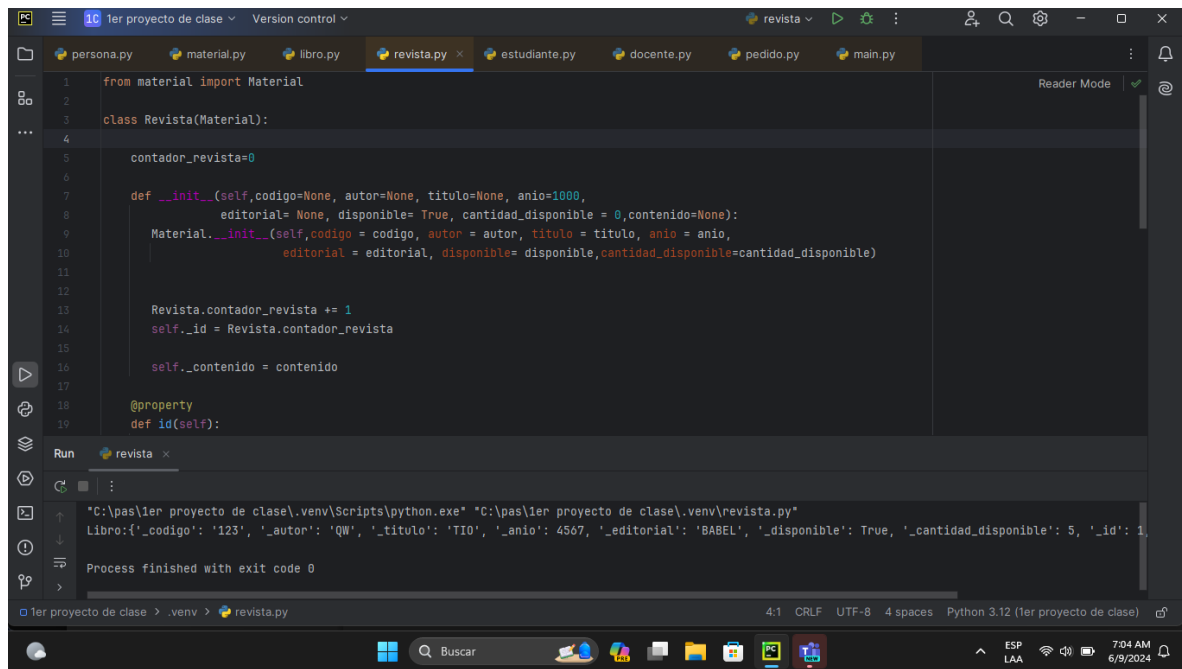
"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\libro.py"

Libro: {'_codigo': 432, '_autor': 'MN', '_titulo': 'primer', '_anio': 2014, '_editorial': 'paco', '_disponible': True, '_cantidad_disponible': 4, '_id': 1, '_tipo_pasta': None}

Process finished with exit code 0

1er proyecto de clase > .venv > libro.py 4:23 CRLF UTF-8 4 spaces Python 3.12 (1er proyecto de clase)

Class revista con su print.



The screenshot shows a code editor with the file 'revista.py' open. The code defines a class 'Revista' that inherits from 'Material'. It includes a class attribute 'contador_revista=0', an '__init__' method that increments the counter and sets attributes, and a property 'id' that returns the instance's ID. The 'Run' button is clicked, and the output console shows the execution path and the printed attributes of a 'Revista' instance.

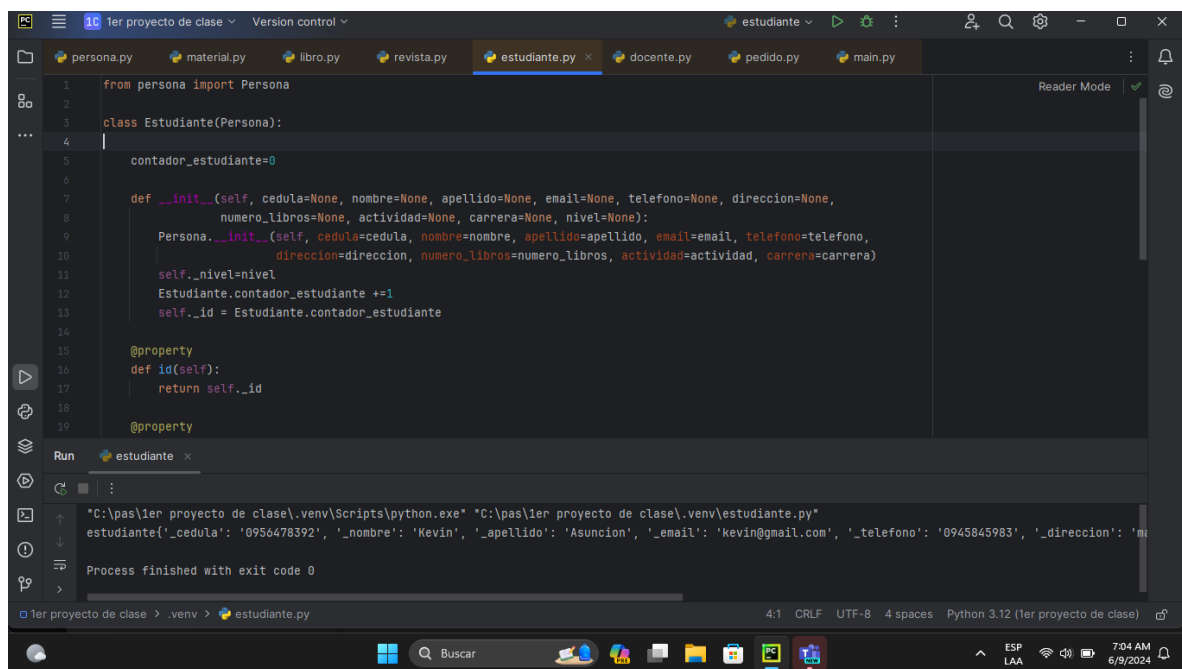
```
1 from material import Material
2
3 class Revista(Material):
4
5     contador_revista=0
6
7     def __init__(self,codigo=None, autor=None, titulo=None, anio=1000,
8                 editorial= None, disponible= True, cantidad_disponible = 0,contenido=None):
9         Material.__init__(self,codigo = codigo, autor = autor, titulo = titulo, anio = anio,
10                          editorial = editorial, disponible= disponible,cantidad_disponible=cantidad_disponible)
11
12
13     Revista.contador_revista += 1
14     self._id = Revista.contador_revista
15
16     self._contenido = contenido
17
18     @property
19     def id(self):
```

Run revista x

"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\revista.py"
Libro:{'_codigo': '123', '_autor': 'QW', '_titulo': 'TIO', '_anio': 4567, '_editorial': 'BABEL', '_disponible': True, '_cantidad_disponible': 5, '_id': 1}

Process finished with exit code 0

Class estudiante con su print.



The screenshot shows a code editor with the file 'estudiante.py' open. The code defines a class 'Estudiante' that inherits from 'Persona'. It includes a class attribute 'contador_estudiante=0', an '__init__' method that increments the counter and sets attributes, and a property 'id' that returns the instance's ID. The 'Run' button is clicked, and the output console shows the execution path and the printed attributes of an 'Estudiante' instance.

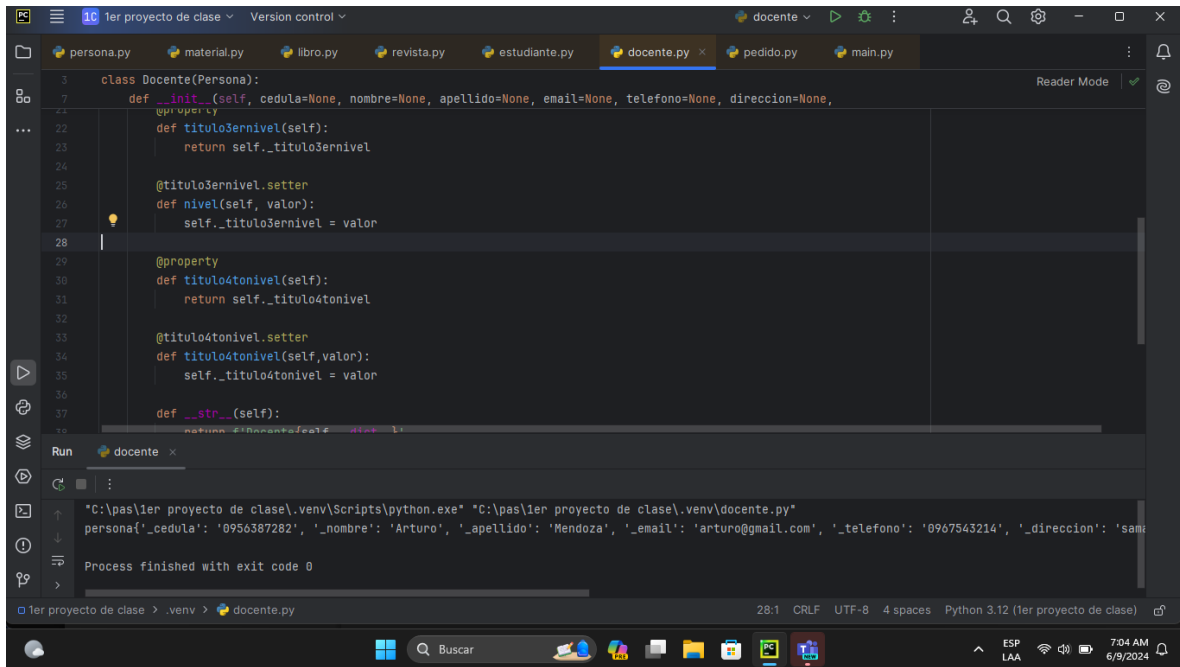
```
1 from persona import Persona
2
3 class Estudiante(Persona):
4
5     contador_estudiante=0
6
7     def __init__(self, cedula=None, nombre=None, apellido=None, email=None, telefono=None, direccion=None,
8                 numero_libros=None, actividad=None, carrera=None, nivel=None):
9         Persona.__init__(self, cedula=cedula, nombre=nombre, apellido=apellido, email=email, telefono=telefono,
10                          direccion=direccion, numero_libros=numero_libros, actividad=actividad, carrera=carrera)
11         self._nivel=nivel
12         Estudiante.contador_estudiante +=1
13         self._id = Estudiante.contador_estudiante
14
15     @property
16     def id(self):
17         return self._id
18
19     @property
```

Run estudiante x

"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\estudiante.py"
estudiante({'cedula': '0956478392', '_nombre': 'Kevin', '_apellido': 'Asuncion', '_email': 'kevin@gmail.com', '_telefono': '0945845983', '_direccion': 'm

Process finished with exit code 0

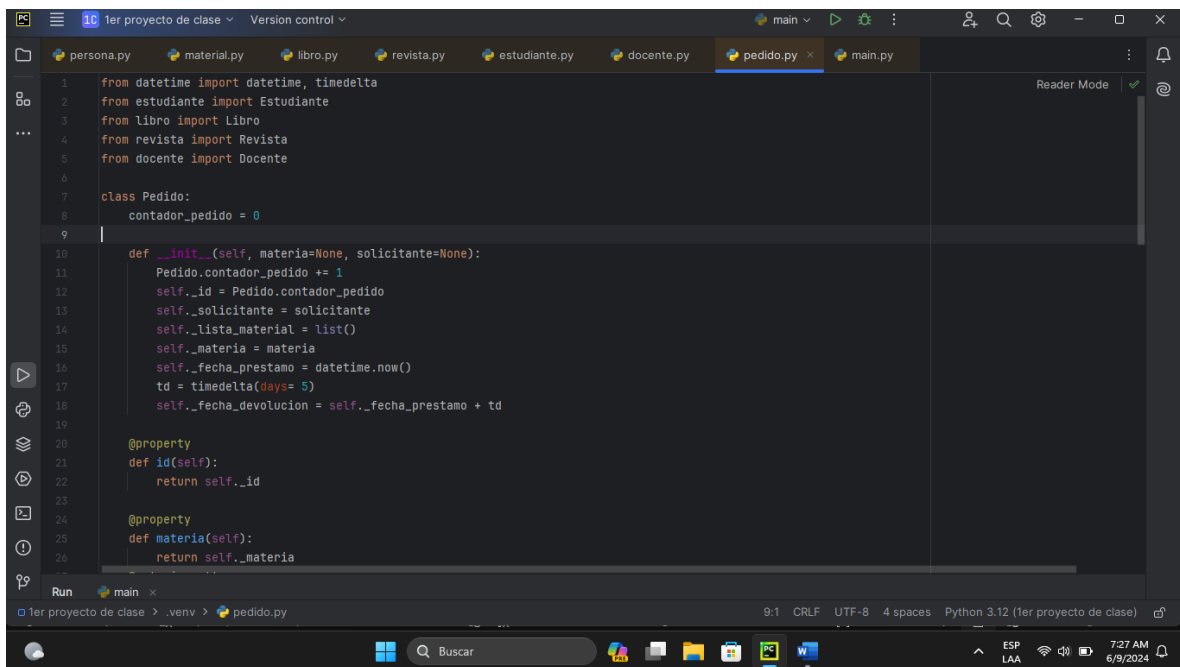
Class docente con su print.



```
3 class Docente(Persona):
7     def __init__(self, cedula=None, nombre=None, apellido=None, email=None, telefono=None, direccion=None,
22     @property
23     def titulo3ernivel(self):
24         return self._titulo3ernivel
25
26     @titulo3ernivel.setter
27     def nivel(self, valor):
28         self._titulo3ernivel = valor
29
30     @property
31     def titulo4tonivel(self):
32         return self._titulo4tonivel
33
34     @titulo4tonivel.setter
35     def titulo4tonivel(self, valor):
36         self._titulo4tonivel = valor
37
38     def __str__(self):
39         return f"Docente {self._nombre} {self._apellido}"

Run docente x
"C:\pas\1er proyecto de clase\.venv\Scripts\python.exe" "C:\pas\1er proyecto de clase\.venv\docente.py"
persona{'_cedula': '0956387282', '_nombre': 'Arturo', '_apellido': 'Mendoza', '_email': 'arturo@gmail.com', '_telefono': '0967543214', '_direccion': 'sami'}
Process finished with exit code 0
```

La Class pedido y main van juntas con main pedimos el print.



```
1 from datetime import datetime, timedelta
2 from estudiante import Estudiante
3 from libro import Libro
4 from revista import Revista
5 from docente import Docente
6
7 class Pedido:
8     contador_pedido = 0
9
10     def __init__(self, materia=None, solicitante=None):
11         Pedido.contador_pedido += 1
12         self._id = Pedido.contador_pedido
13         self._solicitante = solicitante
14         self._lista_materia = list()
15         self._materia = materia
16         self._fecha_prestamo = datetime.now()
17         td = timedelta(days= 5)
18         self._fecha_devolucion = self._fecha_prestamo + td
19
20     @property
21     def id(self):
22         return self._id
23
24     @property
25     def materia(self):
26         return self._materia
```

