PC | File Edit View Navigate Code Refactor Run Tools VCS Window Help   pythonProject - C:\Users\HP\Downloads\docente.py   — ☐ ✕

C: ⟩ Users ⟩ HP ⟩ Downloads ⟩ docente.py

Open Files ▾   docente.py ✕   estudiante.py ✕   libro.py ✕   material.py ✕   pedido.py ✕   persona1.py ✕   revista.py ✕

```python
from persona1 import Persona


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_
        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

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

PC | File Edit View Navigate Code Refactor Run Tools VCS Window Help   pythonProject - C:\Users\HP\Downloads\estudiante.py   — ☐ ✕

C: ⟩ Users ⟩ HP ⟩ Downloads ⟩ estudiante.py

Open Files ▾   docente.py ✕   estudiante.py ✕   libro.py ✕   material.py ✕   pedido.py ✕   persona1.py ✕   revista.py ✕

```python
from persona1 import Persona


class Estudiante(Persona):
    _contador_estudiante = 0

    def __init__(self, cedula, nombre, apellido, email, telefono, direccion, numero_libros, activo, carrera, idES=
                 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
```

```python
    def contador_libro(self):
        return Libro._contador_libro

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

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

    @property
    def tipo_pasta(self):
        return self._tipo_pasta

    @tipo_pasta.setter
    def tipo_pasta(self, nueva_tipo_pasta):
        self._tipo_pasta = nueva_tipo_pasta

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

    def __str__(self):
        return f"Libro(id={self.id}, codigo={self.codigo}, titulo={self.titulo}, autor={self.autor}, disponible={se
```

```python
from abc import ABC, abstractmethod

class Material(ABC):
    def __init__(self, codigo: str = None, autor: str = None, titulo: str = None, anio: int = None, editorial: str
        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):
```

```python
from libro import Libro
from revista import Revista
from estudiante import Estudiante
from docente import Docente
from datetime import datetime

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, fe
        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

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

if __name__ == '__main__'

```python
from abc import ABC, abstractmethod

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__)}'


    @property
    def cedula (self):
        return self._cedula
    @cedula.setter
    def cedula(self, cedula):
        self._cedula = cedula
```

```python
from material import Material


class Revista(Material):
    _contador_revista = 0

    def __init__(self, codigo:str = None, autor:str = None, titulo:str = None, anio:int = None, editorial:str = Nor
        super().__init__(codigo, autor, titulo, anio, editorial, disponible, cantidad_disponible)
        self._id = id
        self._tipo = tipo
        Revista._contador_revista += 1

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

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

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

    @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=

#INTEGRANTES
#MARCOS BUENO, NATALY GARCIA
```

```python
46            return self._fecha_devolucion
47
48        @fecha_devolucion.setter
49        def fecha_devolucion(self, nuevo_fecha_devolucion):
50            self._fecha_devolucion = nuevo_fecha_devolucion
51

     1 usage
52        @property
53        def nombre_solicitante(self):
54            return self._nombre_solicitante
55
56        @nombre_solicitante.setter
57        def nombre_solicitante(self, nuevo_nombre):
58            self._nombre_solicitante = nuevo_nombre
59

     2 usages
60        def pedir_material(self, material):
61            self._lista_material = material
```

Pedido › pedir_material()

Run   🐍 pedido ×

Se ha realizado el préstamo del material Revista(id=None, codigo=211, titulo=National Geographic, autor=TAMBWE, disponible=4, cantidad_disponible=Multiple Disciplines, tipo =No
Fecha de préstamo: 2023-11-27 20:06:23.217656
Se ha devuelto el material Revista(id=None, codigo=211, titulo=National Geographic, autor=TAMBWE, disponible=4, cantidad_disponible=Multiple Disciplines, tipo =None, contador_r
Fecha de devolución: 2023-11-27 20:06:23.217656
Se ha realizado el préstamo del material Libro(id=None, codigo=101, titulo=Alma gemela, autor=Raimon Samsó, disponible=None, cantidad_disponible=None, tipo_pasta=None, contador
Fecha de préstamo: 2023-11-27 20:06:23.217656
Se ha devuelto el material Libro(id=None, codigo=101, titulo=Alma gemela, autor=Raimon Samsó, disponible=None, cantidad_disponible=None, tipo_pasta=None, contador_libro=1) de A
Fecha de devolución: 2023-11-27 20:06:23.217656