

Actividad 1:

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** 21 de nov 09:32 actividad1.py - Act1 - Visual Studio Code
- File Explorer:** Shows a folder structure for ACT1 containing venv_actividad1, actividad1.py, edificios.csv, edificios.json, and log_datos.log.
- Code Editor:** Displays Python code for `actividad1.py`. The code defines a `DataManager` class with methods for updating configuration, adding buildings, and saving to CSV. It uses logging.info to track operations.
- Terminal:** Shows command-line output from running the script:

```
● venv actividad1$usuario@B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act1$ /home/usuario/ACCESO-A-DATOS/Tema6/Act1/actividad1.py
bin/python /home/usuario/ACCESO-A-DATOS/Tema6/Act1/actividad1.py
● venv actividad1$usuario@B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act1$
```
- Output Panel:** Shows log messages from the application:

```
2024-11-21 09:32:43,517 - INFO - Transacción iniciada.
2024-11-21 09:32:43,517 - INFO - Dato agregado: {'nombre': 'Giralda', 'ubicacion': 'España', 'añoConstrucción': '16/06/1650', 'estiloArquitectónico': 'Churrigueresco'}
2024-11-21 09:32:43,517 - INFO - Dato agregado: {'nombre': 'Coliseo Romano', 'ubicacion': 'Roma', 'añoConstrucción': '24/01/1520', 'estiloArquitectónico': 'Pompeyano'}
2024-11-21 09:32:43,517 - INFO - Dato agregado: {'nombre': 'Muro de Berlín', 'ubicacion': 'Alemania', 'añoConstrucción': '01/10/1760', 'estiloArquitectónico': 'Neoclásico'}
2024-11-21 09:32:43,518 - INFO - Transacción confirmada y cambios guardados.
2024-11-21 09:32:43,518 - INFO - Configuración actualizada. Nueva ruta del archivo: edificios.csv
2024-11-21 09:32:43,518 - INFO - Transacción iniciada.
2024-11-21 09:32:43,518 - INFO - Archivo CSV guardado. Versión actual:3
2024-11-21 09:32:43,518 - INFO - Transacción confirmada y cambios guardados.
```

Actividad 2:

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** 25 de nov 14:16 actividad2.py - Act2 - Visual Studio Code
- File Explorer:** Shows a folder structure for ACT2 containing venv_actividad2, actividad2.py, and databasemanager.log.
- Code Editor:** Displays Python code for `actividad2.py`. The code defines a `DatabaseManager` class with methods for connecting to MySQL, creating databases, and creating tables. It uses logging.info for logs.
- Terminal:** Shows command-line output from running the script.
- Output Panel:** Shows log messages from the application.

25 de nov 14:18

● actividad2.py - Act2 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER

ACT2

> venv_actividad2

aktividad2.py

databasemanager.log

```
13 class DatabaseManager:
```

```
38     def crear_edificios(self, nombre, ubicacion, añoConstrucción, estiloArquitectonico):
```

```
39         """Insertar un nuevo edificio en la base de datos."""
```

```
40         try:
```

```
41             cursor = self.connection.cursor()
```

```
42             query = """
```

```
43                 INSERT INTO EdificiosHistoricos (nombre, ubicacion, añoConstrucción, estiloArquitectonico)
```

```
44                 VALUES (%s, %s, %s, %s)
```

```
45             """
```

```
46             cursor.execute(query, (nombre, ubicacion, añoConstrucción, estiloArquitectonico))
```

```
47             logging.info("Edificio '%(nombre)' insertado exitosamente." % {"nombre": nombre})
```

```
48         except Error as e:
```

```
49             logging.error("Error al insertar el edificio '%(nombre)': %(e)s" % {"nombre": nombre, "e": e})
```

```
50     def leer_edificios(self):
```

```
51         """Leer todos los edificios de la base de datos."""
```

```
52         try:
```

```
53             cursor = self.connection.cursor(dictionary=True)
```

```
54             cursor.execute("SELECT * FROM EdificiosHistoricos")
```

```
55             edificios = cursor.fetchall()
```

```
56             logging.info("Edificios recuperados:")
```

```
57             for edificio in edificios:
```

```
58                 logging.info(edificio)
```

```
59             return edificios
```

```
60         except Error as e:
```

```
61             logging.error("Error al leer los edificios: %(e)s" % {"e": e})
```

```
62             return None
```

```
63     def actualizar_edificios(self, nombre, ubicacion, añoConstrucción, estiloArquitectonico):
```

```
64         """Actualizar un edificio en la base de datos."""
```

```
65         try:
```

```
66             cursor = self.connection.cursor()
```

```
67             query = """
```

```
68                 UPDATE EdificiosHistoricos
```

```
69                 SET ubicacion = %s, añoConstrucción = %s, estiloArquitectonico = %s
```

```
70                 WHERE nombre = %s
```

```
71             """
```

```
72             cursor.execute(query, (ubicacion, añoConstrucción, estiloArquitectonico, nombre))
```

```
73             self.connection.commit()
```

```
74             logging.info("Edificio actualizado exitosamente.")
```

Ln 111, Col 68 Spaces:4 UTF-8 LF () Python 3.12.3 ('venv_actividad2':venv) □

25 de nov 14:18

● actividad2.py - Act2 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER

ACT2

> venv_actividad2

aktividad2.py

databasemanager.log

```
13 class DatabaseManager:
```

```
77     def eliminar_edificios(self, nombre):
```

```
78         """Eliminar un edificio de la base de datos."""
```

```
79         try:
```

```
80             cursor = self.connection.cursor()
```

```
81             query = "DELETE FROM EdificiosHistoricos WHERE nombre = %s"
```

```
82             cursor.execute(query, (nombre, ))
```

```
83             self.connection.commit()
```

```
84             logging.info("Edificio eliminado exitosamente.")
```

```
85         except Error as e:
```

```
86             logging.error("Error al eliminar el edificio: %(e)s" % {"e": e})
```

```
87     def iniciar_transaccion(self):
```

```
88         """Iniciar una transacción."""
```

```
89         try:
```

```
90             if self.connection.is_connected():
```

```
91                 self.connection.autocommit = False
```

```
92                 logging.info("Transacción iniciada.")
```

```
93         except Error as e:
```

```
94             logging.error("Error al iniciar la transacción: %(e)s" % {"e": e})
```

```
95     def confirmar_transaccion(self):
```

```
96         """Confirmar (commit) una transacción."""
```

```
97         try:
```

```
98             if self.connection.is_connected():
```

```
99                 self.connection.commit()
```

```
100                self.connection.autocommit = True
```

```
101                logging.info("Transacción confirmada.")
```

```
102        except Error as e:
```

```
103            logging.error("Error al confirmar la transacción: %(e)s" % {"e": e})
```

```
104    def revertir_transaccion(self):
```

```
105        """Revertir (rollback) una transacción."""
```

```
106        try:
```

```
107            if self.connection.is_connected():
```

```
108                self.connection.rollback()
```

```
109                logging.info("Transacción revertida.")
```

```
110        except Error as e:
```

```
111            logging.error("Error al revertir la transacción: %(e)s" % {"e": e})
```

Ejemplo de uso del componente DatabaseManager

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER

ACT2

> venv_actividad2

aktividad2.py

databasemanager.log

```
113 # Ejemplo de uso del componente DatabaseManager
```

```
114 if __name__ == "__main__":
```

```
115     db_manager = DatabaseManager("localhost", "usuario", "usuario", "Idam")
```

```
116     db_manager.conectar()
```

```
117     # Insertar un nuevo edificio
```

```
118     db_manager.crear_edificios("Cristo Redentor", "Rio de Janeiro", 1931, "Renacentista")
```

```
119     db_manager.confirmar_transaccion()
```

```
120     # Leer todos los edificios
```

```
121     db_manager.leer_edificios()
```

```
122     # Actualizar un edificio
```

```
123     db_manager.actualizar_edificios("Cristo Redentor", "Rio de Janeiro", 1931, "Gótico")
```

```
124     # Eliminar un edificio
```

```
125     db_manager.eliminar_edificios("Torre de Pisa")
```

```
126     # Gestionar transacciones
```

```
127     db_manager.iniciar_transaccion()
```

```
128     db_manager.crear_edificios("Pirámides", "Egipto", 2500, "Egipcio")
```

```
129     db_manager.revertir_transaccion() # No se guardará la inserción
```

```
130     # Leer todos los edificios
```

```
131     db_manager.leer_edificios()
```

```
132     db_manager.desconectar()
```

ACT2

venv_actividad2.py

databasemanager.log

```

1 2024-11-25 14:15:01,798 - INFO - Conexión exitosa a la base de datos.
2 2024-11-25 14:15:01,799 - INFO - Edificio 'Cristo Redentor' insertado exitosamente.
3 2024-11-25 14:15:01,802 - INFO - Transacción confirmada.
4 2024-11-25 14:15:01,803 - INFO - Edificios recuperados.
5 2024-11-25 14:15:01,803 - INFO - {'id': 3, 'nombre': 'Taj Majal', 'ubicacion': 'India', 'añoConstrucion': '06/1990', 'estiloArquitectonico': 'Moderno'}
6 2024-11-25 14:15:01,804 - INFO - {'id': 4, 'nombre': 'Torre de Pisa', 'ubicacion': 'Roma', 'añoConstrucion': '15/12/1214', 'estiloArquitectonico': 'Renacentista'}
7 2024-11-25 14:15:01,804 - INFO - {'id': 21, 'nombre': 'Cristo Redentor', 'ubicacion': 'Rio de Janeiro', 'añoConstrucion': '1931', 'estiloArquitectonico': 'Gótico'}
8 2024-11-25 14:15:01,807 - INFO - Edificio actualizado exitosamente.
9 2024-11-25 14:15:01,812 - INFO - Edificio eliminado exitosamente.
10 2024-11-25 14:15:01,812 - INFO - Transacción iniciada.
11 2024-11-25 14:15:01,813 - INFO - Edificio 'Pirámides' insertado exitosamente.
12 2024-11-25 14:15:01,814 - INFO - Transacción revertida.
13 2024-11-25 14:15:01,815 - INFO - Edificios recuperados.
14 2024-11-25 14:15:01,815 - INFO - {'id': 3, 'nombre': 'Taj Majal', 'ubicacion': 'India', 'añoConstrucion': '06/1990', 'estiloArquitectonico': 'Moderno'}
15 2024-11-25 14:15:01,816 - INFO - {'id': 21, 'nombre': 'Cristo Redentor', 'ubicacion': 'Rio de Janeiro', 'añoConstrucion': '1931', 'estiloArquitectonico': 'Gótico'}
16 2024-11-25 14:15:01,817 - INFO - Conexión cerrada.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

venv_actividad2@usuario-B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act2\$ /home/usuario/ACCESO-A-DATOS/Tema6/Act2/actividad2.py

env_actividad2/bin/python /home/usuario/ACCESO-A-DATOS/Tema6/Act2/actividad2.py

2024-11-25 14:15:01,798 - INFO - Conexión exitosa a la base de datos.

2024-11-25 14:15:01,799 - INFO - Edificio 'Cristo Redentor' insertado exitosamente.

2024-11-25 14:15:01,802 - INFO - Transacción confirmada.

2024-11-25 14:15:01,803 - INFO - Edificios recuperados.

2024-11-25 14:15:01,803 - INFO - {'id': 3, 'nombre': 'Taj Majal', 'ubicacion': 'India', 'añoConstrucion': '06/1990', 'estiloArquitectonico': 'Moderno'}

2024-11-25 14:15:01,804 - INFO - {'id': 4, 'nombre': 'Torre de Pisa', 'ubicacion': 'Roma', 'añoConstrucion': '15/12/1214', 'estiloArquitectonico': 'Renacentista'}

2024-11-25 14:15:01,804 - INFO - {'id': 21, 'nombre': 'Cristo Redentor', 'ubicacion': 'Rio de Janeiro', 'añoConstrucion': '1931', 'estiloArquitectonico': 'Gótico'}

2024-11-25 14:15:01,807 - INFO - Edificio actualizado exitosamente.

2024-11-25 14:15:01,812 - INFO - Edificio eliminado exitosamente.

2024-11-25 14:15:01,812 - INFO - Transacción iniciada.

2024-11-25 14:15:01,813 - INFO - Edificio 'Pirámides' insertado exitosamente.

2024-11-25 14:15:01,814 - INFO - Transacción revertida.

2024-11-25 14:15:01,815 - INFO - Edificios recuperados.

2024-11-25 14:15:01,815 - INFO - {'id': 3, 'nombre': 'Taj Majal', 'ubicacion': 'India', 'añoConstrucion': '06/1990', 'estiloArquitectonico': 'Moderno'}

2024-11-25 14:15:01,816 - INFO - {'id': 21, 'nombre': 'Cristo Redentor', 'ubicacion': 'Rio de Janeiro', 'añoConstrucion': '1931', 'estiloArquitectonico': 'Gótico'}

2024-11-25 14:15:01,817 - INFO - Conexión cerrada.

venv_actividad2@usuario-B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act2\$

Actividad 3:

25 de nov 13:37

● actividad3.py - Act3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

actividad3.py > Herramienta

```

1 import logging
2 from peewee import Model, CharField, ForeignKeyField, MySQLDatabase, DoesNotExist
3 # Configuración de logging
4 logging.basicConfig(
5     level=logging.INFO,
6     format='%(asctime)s - %(levelname)s - %(message)s',
7     handlers=[
8         logging.FileHandler("databasemanager_orm.log"),
9         logging.StreamHandler()
10    ]
11 )
12 # Configuración de la base de datos MySQL
13 db = MySQLDatabase(
14     "Idam", # Nombre de la base de datos
15     user="usuario", # Usuario de MySQL
16     password="usuario", # Contraseña de MySQL
17     host="localhost", # Host
18     port=3306 # Puerto por defecto de MySQL
19 )
20 # Modelos de la base de datos
21 class Proveedor(Model):
22     nombre = CharField(unique=True)
23     direccion = CharField()
24     class Meta:
25         database = db
26
27 class Herramienta(Model):
28     nombre = CharField()
29     tipo = CharField()
30     marca = CharField()
31     uso = CharField()
32     material = CharField()
33     proveedor = ForeignKeyField(Proveedor, backref='herramientas')
34     class Meta:
35         database = db
36
37 # Componente DatabaseManagerORM
38

```

Ln 33, Col 67 Spaces:4 UTF-8 LF {} Python 3.12.3 (venv_actividad3:venv)

25 de nov 13:37

● actividad3.py - Act3 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

```
Welcome actividad3.py
```

```
aktividad3.py > DatabaseManagerORM > eliminar_proveedor
```

```
30
31
32
33
34
35 # Componente DatabaseManagerORM
36
37 class DatabaseManagerORM:
38     def __init__(self):
39         self.db = db
40     def conectar(self):
41         """Conecta la base de datos y crea las tablas."""
42         self.db.connect()
43         self.db.create_tables([Proveedor, Herramienta])
44         logging.info("Conexión establecida y tablas creadas.")
45     def desconectar(self):
46         """Cierra la conexión a la base de datos."""
47         if not self.db.is_closed():
48             self.db.close()
49             logging.info("Conexión cerrada.")
50     def iniciar_transaccion(self):
51         """Inicia una transacción."""
52         self.db.begin()
53         logging.info("Transacción iniciada.")
54     def confirmar_transaccion(self):
55         """Confirma (commit) una transacción."""
56         self.db.commit()
57         logging.info("Transacción confirmada.")
58     def revertir_transaccion(self):
59         """Revierte (rollback) una transacción."""
60         self.db.rollback()
61         logging.info("Transacción revertida.")
62     def crear_proveedor(self, nombre, dirección):
63         """Inserta un nuevo proveedor."""
64         proveedor = Proveedor.create(nombre=nombre, dirección=dirección)
65         logging.info(f"Proveedor creado: {proveedor.nombre} - {proveedor.dirección}")
66         return proveedor
67     def crear_herramienta(self, nombre, tipo, marca, uso, material, proveedor_nombre):
68         """Inserta una nueva herramienta."""
69         proveedor = Proveedor.get_or_none(Proveedor.nombre == proveedor_nombre)
70         if not proveedor:
71             raise ValueError(f"No existe un proveedor con el nombre {proveedor_nombre}")
72         herramienta = Herramienta.create(
73             nombre=nombre, tipo=tipo, marca=marca, uso=uso, material=material, proveedor=proveedor
74         )
75         logging.info(f"Herramienta creada: {herramienta.nombre} - {herramienta.tipo}")
76         return herramienta
77     def leer_herramientas(self):
78         """Lee todas las herramientas."""
79         herramientas = Herramienta.select()
80         logging.info("Leyendo herramientas:")
81         for herramienta in herramientas:
82             logging.info(f"{herramienta.nombre} - {herramienta.tipo} ({herramienta.proveedor.nombre})")
83     def actualizar_herramienta(self, nombre, tipo, marca, uso, material):
84         """Actualiza una herramienta en la base de datos."""
85         herramienta = Herramienta.get(Herramienta.nombre == nombre)
86         herramienta.tipo = tipo
87         herramienta.marca = marca
88         herramienta.uso = uso
89         herramienta.material = material
90         herramienta.save()
91         logging.info(f"Herramienta {nombre} actualizada exitosamente.")
92     def actualizar_proveedor(self, nombre, dirección):
93         """Actualiza un proveedor en la base de datos."""
94         proveedor = Proveedor.get(Proveedor.nombre == nombre)
95         proveedor.dirección = dirección
96         proveedor.save()
97         logging.info(f"Proveedor {nombre} actualizado exitosamente.")
98     def eliminar_proveedor(self, nombre):
99         """Elimina un proveedor de la base de datos."""
100        proveedor = Proveedor.get(Proveedor.nombre == nombre)
101        proveedor.delete()
102        logging.info(f"Proveedor {nombre} eliminado exitosamente.")
```

Ln 121, Col 79 Spaces:4 UTF-8 LF {} Python 3.12.3 (venv_actividad3:venv)

25 de nov 13:38

● actividad3.py - Act3 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

```
Welcome actividad3.py
```

```
aktividad3.py > DatabaseManagerORM > eliminar_proveedor
```

```
38 class DatabaseManagerORM:
39     def crear_herramienta(self, nombre, tipo, marca, uso, material, proveedor_nombre):
40         """Inserta una nueva herramienta."""
41         proveedor = Proveedor.get_or_none(Proveedor.nombre == proveedor_nombre)
42         if not proveedor:
43             raise ValueError(f"No existe un proveedor con el nombre {proveedor_nombre}")
44         herramienta = Herramienta.create(
45             nombre=nombre, tipo=tipo, marca=marca, uso=uso, material=material, proveedor=proveedor
46         )
47         logging.info(f"Herramienta creada: {herramienta.nombre} - {herramienta.tipo}")
48         return herramienta
49     def leer_herramientas(self):
50         """Lee todas las herramientas."""
51         herramientas = Herramienta.select()
52         logging.info("Leyendo herramientas:")
53         for herramienta in herramientas:
54             logging.info(f"{herramienta.nombre} - {herramienta.tipo} ({herramienta.proveedor.nombre})")
55     def actualizar_herramienta(self, nombre, tipo, marca, uso, material):
56         """Actualiza una herramienta en la base de datos."""
57         herramienta = Herramienta.get(Herramienta.nombre == nombre)
58         herramienta.tipo = tipo
59         herramienta.marca = marca
60         herramienta.uso = uso
61         herramienta.material = material
62         herramienta.save()
63         logging.info(f"Herramienta {nombre} actualizada exitosamente.")
64     def actualizar_proveedor(self, nombre, dirección):
65         """Actualiza un proveedor en la base de datos."""
66         proveedor = Proveedor.get(Proveedor.nombre == nombre)
67         proveedor.dirección = dirección
68         proveedor.save()
69         logging.info(f"Proveedor {nombre} actualizado exitosamente.")
70     def eliminar_proveedor(self, nombre):
71         """Elimina un proveedor de la base de datos."""
72         proveedor = Proveedor.get(Proveedor.nombre == nombre)
73         proveedor.delete()
74         logging.info(f"Proveedor {nombre} eliminado exitosamente.")
```

Ln 121, Col 79 Spaces:4 UTF-8 LF {} Python 3.12.3 (venv_actividad3:venv)

25 de nov 13:38

● actividad3.py - Act3 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
Welcome actividad3.py ●
actividad3.py > DatabaseManagerORM > eliminar_proveedor
38 class DatabaseManagerORM:
    105     logging.error(f"No se encontró el proveedor con nombre {nombre}.")
    106     def eliminar_herramienta(self, nombre):
    107         """Eliminar una herramienta de la base de datos."""
    108         try:
    109             herramienta = Herramienta.get(Herramienta.nombre == nombre)
    110             herramienta.delete_instance()
    111             logging.info(f"Herramienta {nombre} eliminada exitosamente.")
    112         except DoesNotExist:
    113             logging.error(f"No se encontró la herramienta con nombre {nombre}.")
    114         def eliminar_proveedor(self, nombre):
    115             """Eliminar un proveedor de la base de datos."""
    116             try:
    117                 proveedor = Proveedor.get(Proveedor.nombre == nombre)
    118                 proveedor.delete_instance()
    119                 logging.info(f"Proveedor {nombre} eliminado exitosamente.")
    120             except DoesNotExist:
    121                 logging.error(f"No se encontró el proveedor con nombre {nombre}.")
    122             def consultar_herramientas_proveedor(self, proveedor_nombre):
    123                 """Consulta herramientas asociadas a un proveedor."""
    124                 proveedor = Proveedor.get_or_none(Proveedor.nombre == proveedor_nombre)
    125                 if not proveedor:
    126                     logging.error(f"No existe un proveedor con el nombre {proveedor_nombre}")
    127                     return
    128                 herramientas = Herramienta.select().where(Herramienta.proveedor == proveedor)
    129                 logging.info(f"Herramientas asociadas a {proveedor.nombre}:")
    130                 for herramienta in herramientas:
    131                     logging.info(f"{{herramienta.nombre}} - {{herramienta.tipo}}")

    134     # Ejemplo de uso
    135     database = DatabaseManagerORM()
    136     database.conectar()
    137     print("\nGestión de Proveedores:")
    138     database.iniciar_transaccion()
    139     database.crear_proveedor("ProveedorA", "123-456-789")
    140     database.crear_proveedor("ProveedorB", "987-654-321")
    141     database.confirmar_transaccion()

Ln 121, Col 79 Spaces:4 UTF-8 LF () Python 3.12.3 ('venv_actividad3':venv) □
```

25 de nov 13:39

● actividad3.py - Act3 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
Welcome actividad3.py ●
actividad3.py > DatabaseManagerORM > eliminar_proveedor
133
134     # Ejemplo de uso
135     database = DatabaseManagerORM()
136     database.conectar()
137     print("\nGestión de Proveedores:")
138     database.iniciar_transaccion()
139     database.crear_proveedor("ProveedorA", "123-456-789")
140     database.crear_proveedor("ProveedorB", "987-654-321")
141     database.confirmar_transaccion()
142     print("\nCambio de dato por mi DNI 30284233H al ProveedorA:")
143     database.iniciar_transaccion()
144     database.actualizar_proveedor("ProveedorA", "30284233")
145     database.confirmar_transaccion()
146     print("Eliminar ProveedorB:")
147     database.iniciar_transaccion()
148     database.eliminar_proveedor("ProveedorB")
149     database.confirmar_transaccion()
150     print("\nGestión de Herramientas:")
151     database.iniciar_transaccion()
152     database.crear_herramienta("Martillo", "Manual", "Facom", "Percusión", "Acero", "ProveedorA")
153     database.crear_herramienta("Taladro", "Eléctrico", "Facom", "Percusión", "Acero", "ProveedorA")
154     database.confirmar_transaccion()
155     print("\nHerramientas asociadas al ProveedorA:")
156     database.consultar_herramientas_proveedor("ProveedorA")
157     database.iniciar_transaccion()
158     print("\nActualizar herramienta Martillo a tipo reforzado:")
159     database.actualizar_herramienta("Martillo", "Reforzado", "Facom", "Percusión", "Acero")
160     database.confirmar_transaccion()
161     print("\nEliminar herramienta Taladro:")
162     database.iniciar_transaccion()
163     database.eliminar_herramienta("Taladro")
164     database.confirmar_transaccion()
165     database.desconectar()

Ln 121, Col 79 Spaces:4 UTF-8 LF () Python 3.12.3 ('venv_actividad3':venv) □
```

25 de nov 13:42

actividad3.py - Act3 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Welcome actividad3.py
aktividad3.py > ...
l11 database.eliminar_proveedor("ProveedorB")
2024-11-25 13:42:43,346 - INFO - Conexión establecida y tablas creadas.

Gestión de Proveedores:
2024-11-25 13:42:43,347 - INFO - Transacción iniciada.
2024-11-25 13:42:43,348 - INFO - Proveedor creado: ProveedorA - 123-456-789
2024-11-25 13:42:43,348 - INFO - Proveedor creado: ProveedorB - 987-654-321
2024-11-25 13:42:43,351 - INFO - Transacción confirmada.

Cambio de dato por mí DNI 30284233H al ProveedorA:
2024-11-25 13:42:43,351 - INFO - Transacción iniciada.
2024-11-25 13:42:43,353 - INFO - Proveedor ProveedorA actualizado exitosamente.
2024-11-25 13:42:43,357 - INFO - Transacción confirmada.

Eliminar ProveedorB:
2024-11-25 13:42:43,357 - INFO - Transacción iniciada.
2024-11-25 13:42:43,358 - INFO - Proveedor ProveedorB eliminado exitosamente.
2024-11-25 13:42:43,361 - INFO - Transacción confirmada.

Gestión de Herramientas:
2024-11-25 13:42:43,362 - INFO - Transacción iniciada.
2024-11-25 13:42:43,363 - INFO - Herramienta creada: Martillo - Manual
2024-11-25 13:42:43,365 - INFO - Herramienta creada: Taladro - Eléctrico
2024-11-25 13:42:43,369 - INFO - Transacción confirmada.

Herramientas asociadas al ProveedorA:
2024-11-25 13:42:43,370 - INFO - Herramientas asociadas a ProveedorA
2024-11-25 13:42:43,372 - INFO - Martillo - Manual
2024-11-25 13:42:43,372 - INFO - Taladro - Eléctrico
2024-11-25 13:42:43,372 - INFO - Transacción iniciada.

Actualizar herramienta Martillo a tipo reforzado:
2024-11-25 13:42:43,374 - INFO - Herramienta Martillo actualizada exitosamente.
2024-11-25 13:42:43,377 - INFO - Transacción confirmada.

Eliminar herramienta Taladro:
2024-11-25 13:42:43,377 - INFO - Transacción iniciada.
2024-11-25 13:42:43,379 - INFO - Herramienta Taladro eliminada exitosamente.
2024-11-25 13:42:43,381 - INFO - Transacción confirmada.
2024-11-25 13:42:43,381 - INFO - Conexión cerrada.

venv actividad3$ usuario@usuario-B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act3$
```

Ln 167, Col 33 Spaces: 4 UTF-8 () Python 3.12.3 (venv_actividad3):venv

EXPLORER actividad3.py databasemanager_orm.log

databasemanager_orm.log

```

1 2024-11-25 13:44:10,252 - INFO - Conexión establecida y tablas creadas.
2 2024-11-25 13:44:10,252 - INFO - Transacción iniciada.
3 2024-11-25 13:44:10,253 - INFO - Proveedor creado: ProveedorA - 123-456-789
4 2024-11-25 13:44:10,254 - INFO - Proveedor creado: ProveedorB - 987-654-321
5 2024-11-25 13:44:10,257 - INFO - Transacción confirmada.
6 2024-11-25 13:44:10,257 - INFO - Transacción iniciada.
7 2024-11-25 13:44:10,259 - INFO - Proveedor ProveedorA actualizado exitosamente.
8 2024-11-25 13:44:10,264 - INFO - Transacción confirmada.
9 2024-11-25 13:44:10,265 - INFO - Transacción iniciada.
10 2024-11-25 13:44:10,266 - INFO - Proveedor ProveedorB eliminado exitosamente.
11 2024-11-25 13:44:10,269 - INFO - Transacción confirmada.
12 2024-11-25 13:44:10,269 - INFO - Transacción iniciada.
13 2024-11-25 13:44:10,271 - INFO - Herramienta creada: Martillo - Manual
14 2024-11-25 13:44:10,272 - INFO - Herramienta creada: Taladro - Eléctrico
15 2024-11-25 13:44:10,278 - INFO - Transacción confirmada.
16 2024-11-25 13:44:10,280 - INFO - Herramientas asociadas a ProveedorA:
17 2024-11-25 13:44:10,282 - INFO - Martillo - Manual
18 2024-11-25 13:44:10,282 - INFO - Taladro - Eléctrico
19 2024-11-25 13:44:10,282 - INFO - Transacción iniciada.
20 2024-11-25 13:44:10,286 - INFO - Herramienta Martillo actualizada exitosamente.
21 2024-11-25 13:44:10,292 - INFO - Transacción confirmada.
22 2024-11-25 13:44:10,292 - INFO - Transacción iniciada.
23 2024-11-25 13:44:10,296 - INFO - Herramienta Taladro eliminada exitosamente.
24 2024-11-25 13:44:10,300 - INFO - Transacción confirmada.
25 2024-11-25 13:44:10,301 - INFO - Conexión cerrada.
```

Actividad 4:

```

● venv actividad4$ usuario@usuario-B250M-DS3H:~/ACCESO-A-DATOS/Tema6/Act4$ /home/usuario/ACCESO-A-DATOS/Tema6/Act4/venv_actividad4/bin/python /home/usuario/ACCESO-A-DATOS/Tema6/Act4/actividad4.py
2024-12-02 12:36:02,076 - INFO - Conectado a MongoDB: idam.edificios
2024-12-02 12:36:02,076 - INFO - Transacción iniciada.
2024-12-02 12:36:02,078 - INFO - Documento insertado con ID: 674d9ba2401e871572d0c527
2024-12-02 12:36:02,078 - INFO - Documento insertado con ID: 674d9ba2401e871572d0c528
2024-12-02 12:36:02,078 - INFO - Transacción confirmada.
2024-12-02 12:36:02,079 - INFO - Documentos recuperados: 3
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba3028e489b917a327')}, 'nombre': 'Louvre', 'ubicacion': 'Francia', 'año Construcción': '01/12/1300', 'estiloArquitectónico': 'Moderno'}
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c527')}, 'nombre': 'Louvre', 'ubicacion': 'Francia', 'año Construcción': '12/01/1100', 'estiloArquitectónico': 'Moderno'}
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c528')}, 'nombre': 'Big Ben', 'ubicacion': 'Inglaterra', 'año Construcción': '30/09/1200', 'estiloArquitectónico': 'Contemporáneo'}
2024-12-02 12:36:02,079 - INFO - Transacción iniciada.
2024-12-02 12:36:02,079 - WARNING - No se encontró documento para actualizar: {'nombre': 'Louvre'}
2024-12-02 12:36:02,080 - INFO - Transacción confirmada.
2024-12-02 12:36:02,080 - INFO - Transacción iniciada.
2024-12-02 12:36:02,080 - INFO - Documento eliminado: {'nombre': 'Big Ben'}
2024-12-02 12:36:02,080 - INFO - Transacción confirmada.
2024-12-02 12:36:02,081 - INFO - Documentos recuperados: 2
2024-12-02 12:36:02,081 - INFO - {'_id': ObjectId('674d9ba3028e489b917a327')}, 'nombre': 'Louvre', 'ubicacion': 'Francia', 'año Construcción': '01/12/1300', 'estiloArquitectónico': 'Moderno'}
2024-12-02 12:36:02,081 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c527')}, 'nombre': 'Louvre', 'ubicacion': 'Francia', 'año Construcción': '12/01/1100', 'estiloArquitectónico': 'Moderno'}
2024-12-02 12:36:02,082 - INFO - Conexión a MongoDB cerrada.
venv actividad4$
```

The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** "actividad4.py - Act4 - Visual Studio Code" and the date "2 de dic 12:37".
- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer View:** Shows a file tree with "ACT4" expanded, containing "venv_actividad4", "actividad4.py", and "databasemanager_documental.log".
- Code Editor:** The main area displays Python code for a "DatabaseManagerDocumental" class using PyMongo to interact with a MongoDB database. The code includes methods for connecting and disconnecting from the database.
- Right Panel:** A sidebar on the right shows a preview of the "databasemanager_documental.log" file, which contains several log entries.
- Sidebar Icons:** Various icons for file types like JSON, YAML, and Markdown, as well as navigation and search icons.
- Bottom Status Bar:** "Ln 14, Col 1" and "Spaces: 4" and "UTF-8".

The screenshot shows a Visual Studio Code interface with the following details:

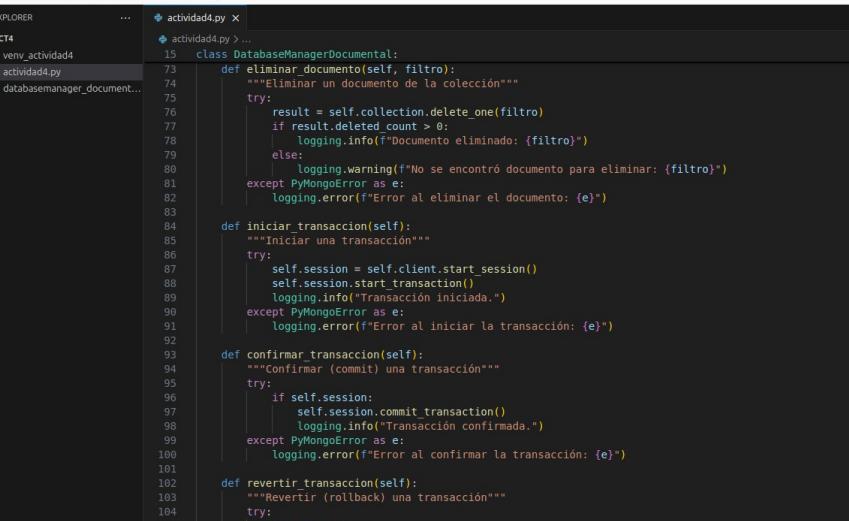
- Title Bar:** 2 de dic 12:37, actividad4.py - Act4 - Visual Studio Code
- File Explorer (Left Side):** Shows a project structure with a folder named ACT4 containing a venv_aktividad4 folder and an actividad4.py file. The actividad4.py file is currently selected.
- Code Editor (Main Area):** Displays Python code for a DatabaseManagerDocumental class. The code includes methods for creating, reading, updating, and deleting documents from a MongoDB collection. It uses logging.info for informational messages and logging.error for errors.
- Right Panel:** Shows a vertical stack of code snippets or tabs, likely related to the current file or project.

```
class DatabaseManagerDocumental:
    def crear_documento(self, documento):
        """Insertar un nuevo documento en la colección"""
        try:
            result = self.collection.insert_one(documento)
            logging.info(f"Documento insertado con ID: {result.inserted_id}")
            return result.inserted_id
        except PyMongoError as e:
            logging.error(f"Error al insertar el documento: {e}")

    def leer_documentos(self, filtro={}):
        """Leer documentos de la colección según un filtro"""
        try:
            documentos = list(self.collection.find(filtro))
            logging.info(f"Documentos recuperados: {len(documentos)}")
            for doc in documentos:
                logging.info(doc)
            return documentos
        except PyMongoError as e:
            logging.error(f"Error al leer los documentos: {e}")
            return []

    def actualizar_documento(self, filtro, actualizacion):
        """Actualizar un documento en la colección"""
        try:
            result = self.collection.update_one(filtro, {"$set": actualizacion})
            if result.modified_count > 0:
                logging.info(f"Documento actualizado: {filtro}")
            else:
                logging.warning(f"No se encontró documento para actualizar: {filtro}")
        except PyMongoError as e:
            logging.error(f"Error al actualizar el documento: {e}")

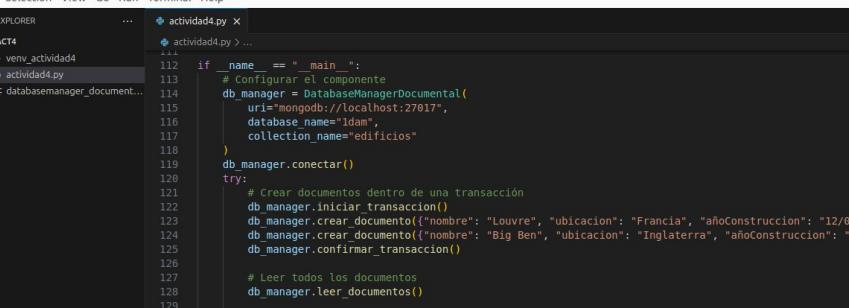
    def eliminar_documento(self, filtro):
        """Eliminar un documento de la colección"""
        try:
            result = self.collection.delete_one(filtro)
            if result.deleted_count > 0:
```



The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** actividad4.py - Act4 - Visual Studio Code
- File Menu:** File Edit Selection View Go Run Terminal Help
- Sidebar:** EXPLORER, ACT4, OUTLINE, TIMELINE.
- Code Editor:** Python code for a DatabaseManagerDocumental class:

```
15  class DatabaseManagerDocumental:  
16      """ Eliminar un documento de la colección """  
17      def eliminar_documento(self, filtro):  
18          try:  
19              result = self.collection.delete_one(filtro)  
20              if result.deleted_count > 0:  
21                  logging.info("Documento eliminado: {filtro}")  
22              else:  
23                  logging.warning("No se encontró documento para eliminar: {filtro}")  
24          except PyMongoError as e:  
25              logging.error(f"Error al eliminar el documento: {e}")  
26  
27      def iniciar_transaccion(self):  
28          """ Iniciar una transacción """  
29          try:  
30              self.session = self.client.start_session()  
31              self.session.start_transaction()  
32              logging.info("Transacción iniciada.")  
33          except PyMongoError as e:  
34              logging.error(f"Error al iniciar la transacción: {e}")  
35  
36      def confirmar_transaccion(self):  
37          """ Confirmar (commit) una transacción """  
38          try:  
39              if self.session:  
40                  self.session.commit_transaction()  
41                  logging.info("Transacción confirmada.")  
42          except PyMongoError as e:  
43              logging.error(f"Error al confirmar la transacción: {e}")  
44  
45      def revertir_transaccion(self):  
46          """ Revertir (rollback) una transacción """  
47          try:  
48              if self.session:  
49                  self.session.abort_transaction()  
50                  logging.info("Transacción revertida.")  
51          except PyMongoError as e:  
52              logging.error(f"Error al revertir la transacción: {e}")
```
- Right Panel:** Includes a file tree, a search bar, and a status bar at the bottom.

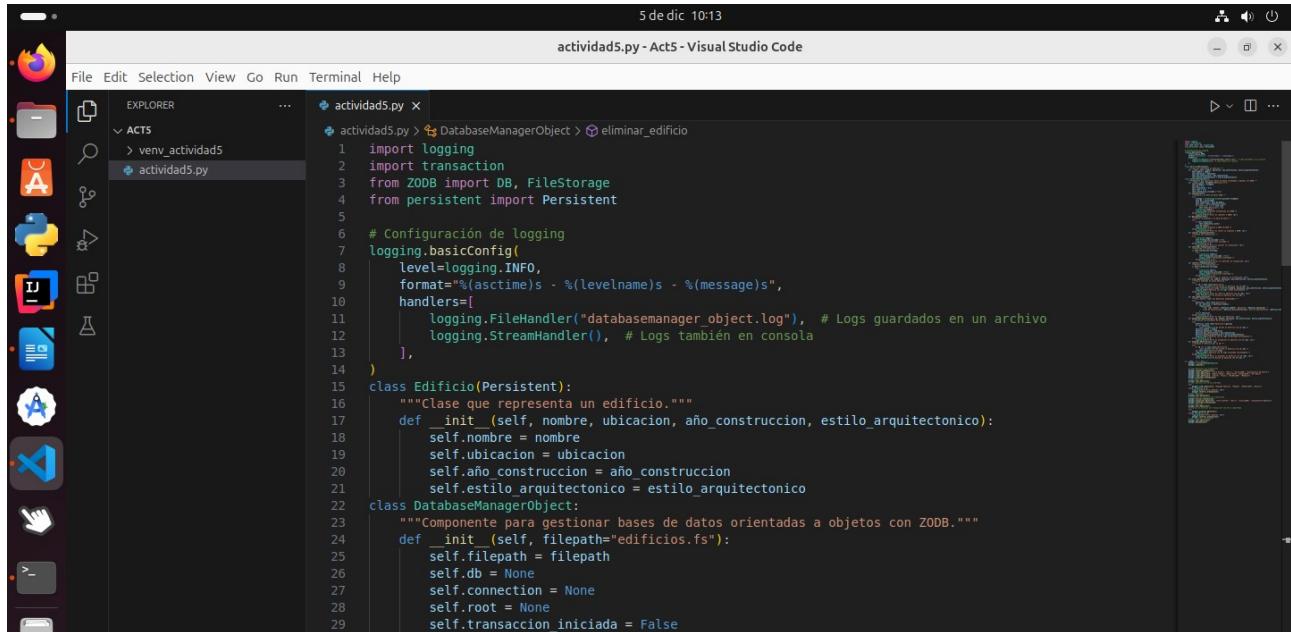


The screenshot shows a Visual Studio Code interface with the following details:

- Title Bar:** actividad4.py - Act4 - Visual Studio Code
- File Explorer:** Shows a project structure under ACT4:
 - actividad4.py
 - databaseManagerDocument...
- Code Editor:** The main editor pane displays Python code for managing a database. The code includes imports for `DatabaseManager` and `MongoDBClient`, and uses them to interact with a MongoDB database named 'ldam' containing a collection 'edificios'. It demonstrates creating documents, reading all documents, updating a document, deleting a document, and finally disconnecting from the database.
- Terminal:** Not visible in the screenshot.
- Status Bar:** Shows the current file is Ln 14, Col 1, with settings for Spaces: 4, UTF-8, LF, Python, and a path to 3.12.3 (venv_actividad4):venv.

```
2024-12-02 12:36:02,076 - INFO - Conectado a MongoDB: 1dam.edificios
2024-12-02 12:36:02,076 - INFO - Transacción iniciada.
2024-12-02 12:36:02,078 - INFO - Documento insertado con ID: 674d9ba2401e871572d0c527
2024-12-02 12:36:02,078 - INFO - Documento insertado con ID: 674d9ba2401e871572d0c528
2024-12-02 12:36:02,078 - INFO - Transacción confirmada.
2024-12-02 12:36:02,079 - INFO - Documentos recuperados: 3
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba3a3028e489b917a327'), 'nombre': 'Louvre', 'ubicacion': 'París'}
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c527'), 'nombre': 'Louvre', 'ubicacion': 'París'}
2024-12-02 12:36:02,079 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c528'), 'nombre': 'Big Ben', 'ubicacion': 'Londres'}
2024-12-02 12:36:02,079 - INFO - Transacción iniciada.
2024-12-02 12:36:02,079 - WARNING - No se encontró documento para actualizar: {'nombre': 'Louvre'}
2024-12-02 12:36:02,080 - INFO - Transacción confirmada.
2024-12-02 12:36:02,080 - INFO - Transacción iniciada.
2024-12-02 12:36:02,080 - INFO - Documento eliminado: {'nombre': 'Big Ben'}
2024-12-02 12:36:02,080 - INFO - Transacción confirmada.
2024-12-02 12:36:02,081 - INFO - Documentos recuperados: 2
2024-12-02 12:36:02,081 - INFO - {'_id': ObjectId('674d9ba3a3028e489b917a327'), 'nombre': 'Louvre', 'ubicacion': 'París'}
2024-12-02 12:36:02,081 - INFO - {'_id': ObjectId('674d9ba2401e871572d0c527'), 'nombre': 'Louvre', 'ubicacion': 'París'}
2024-12-02 12:36:02,082 - INFO - Conexión a MongoDB cerrada.
```

Actividad 5:

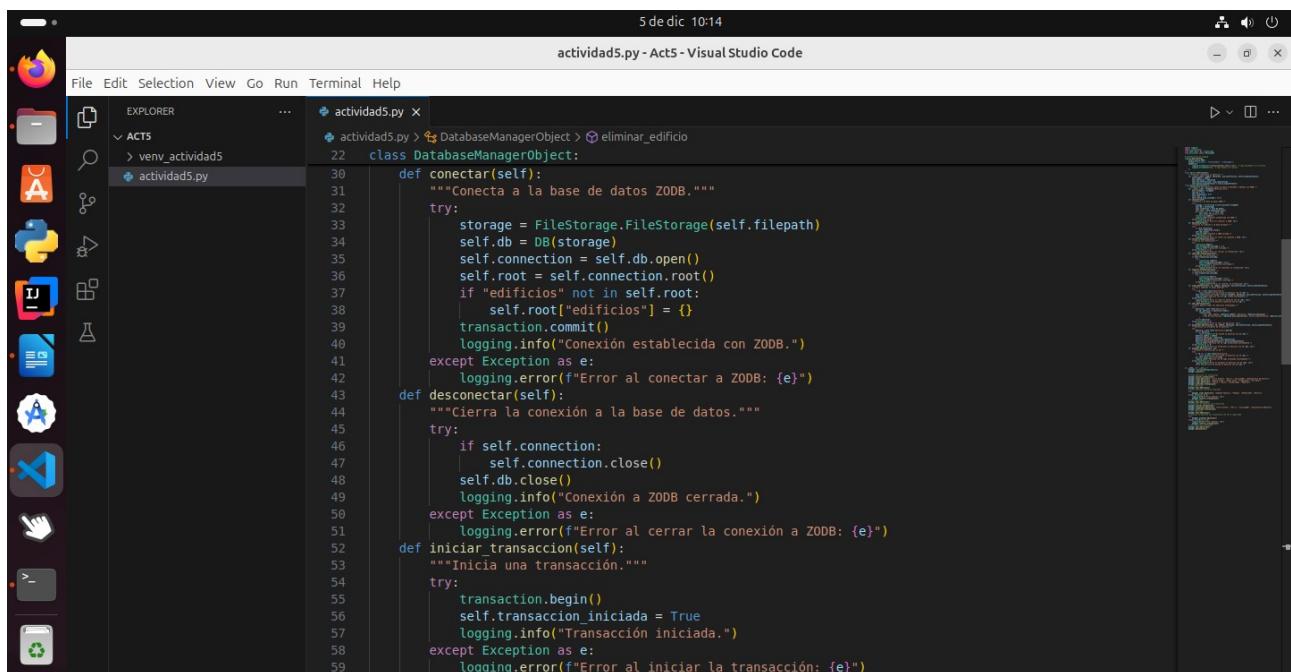


5 de dic 10:13 actividad5.py - Act5 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER ACTS venv_actividad5 actividad5.py

```
1 import logging
2 import transaction
3 from ZODB import DB, FileStorage
4 from persistent import Persistent
5
6 # Configuración de logging
7 logging.basicConfig(
8     level=logging.INFO,
9     format="%(asctime)s - %(levelname)s - %(message)s",
10    handlers=[
11        logging.FileHandler("databasemanager_object.log"), # Logs guardados en un archivo
12        logging.StreamHandler(), # Logs también en consola
13    ],
14)
15 class Edificio(Persistent):
16     """Clase que representa un edificio."""
17     def __init__(self, nombre, ubicacion, año_construcción, estilo_arquitectónico):
18         self.nombre = nombre
19         self.ubicacion = ubicacion
20         self.año_construcción = año_construcción
21         self.estilo_arquitectónico = estilo_arquitectónico
22 class DatabaseManagerObject:
23     """Componente para gestionar bases de datos orientadas a objetos con ZODB."""
24     def __init__(self, filepath="edificios.fs"):
25         self.filepath = filepath
26         self.db = None
27         self.connection = None
28         self.root = None
29         self.transacción_iniciada = False
```



5 de dic 10:14 actividad5.py - Act5 - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help
```

EXPLORER ACTS venv_actividad5 actividad5.py

```
22 class DatabaseManagerObject:
23     def conectar(self):
24         """Conecta a la base de datos ZODB."""
25         try:
26             storage = FileStorage.FileStorage(self.filepath)
27             self.db = DB(storage)
28             self.connection = self.db.open()
29             self.root = self.connection.root()
30             if "edificios" not in self.root:
31                 self.root["edificios"] = {}
32             transaction.commit()
33             logging.info("Conexión establecida con ZODB.")
34         except Exception as e:
35             logging.error(f"Error al conectar a ZODB: {e}")
36     def desconectar(self):
37         """Cierra la conexión a la base de datos."""
38         try:
39             if self.connection:
40                 self.connection.close()
41             self.db.close()
42             logging.info("Conexión a ZODB cerrada.")
43         except Exception as e:
44             logging.error(f"Error al cerrar la conexión a ZODB: {e}")
45     def iniciar_transacción(self):
46         """Inicia una transacción."""
47         try:
48             transaction.begin()
49             self.transacción_iniciada = True
50             logging.info("Transacción iniciada.")
51         except Exception as e:
52             logging.error(f"Error al iniciar la transacción: {e}")
```

```
5 de dic 10:14 actividad5.py - Act5 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER ACTS actividad5.py
aktividad5.py > DatabaseManagerObject > eliminar_edificio
22 class DatabaseManagerObject:
23     def confirmar_transaccion(self):
24         """Confirma la transacción."""
25         if self.transaccion_iniciada:
26             try:
27                 transaction.commit()
28                 self.transaccion_iniciada = False
29                 logging.info("Transacción confirmada.")
30             except Exception as e:
31                 logging.error(f"Error al confirmar la transacción: {e}")
32         def revertir_transaccion(self):
33             """Revierte la transacción."""
34             if self.transaccion_iniciada:
35                 try:
36                     transaction.abort()
37                     self.transaccion_iniciada = False
38                     logging.info("Transacción revertida.")
39                 except Exception as e:
40                     logging.error(f"Error al revertir la transacción: {e}")
41     def crear_edificio(self, id, nombre, ubicacion, año_construcción, estilo_arquitectónico):
42         """Crea y almacena un nuevo edificio."""
43         try:
44             if id in self.root["edificios"]:
45                 raise ValueError(f"Ya existe un edificio con ID {id}.")
46             self.root["edificios"][id] = Edificio(nombre, ubicacion, año_construcción, estilo_arquitectónico)
47             logging.info(f"Edificio con ID {id} creado exitosamente.")
48         except Exception as e:
49             logging.error(f"Error al crear el edificio con ID {id}: {e}")
50             raise ValueError(f"Ya existe un edificio con ID {id}.")
```

```
5 de dic 10:14 actividad5.py - Act5 - Visual Studio Code

File Edit Selection View Go Run Terminal Help
EXPLORER ACTS actividad5.py
aktividad5.py > DatabaseManagerObject > eliminar_edificio
88     def leer_edificios(self):
89         """Lee y muestra todos los edificios almacenados."""
90         try:
91             edificios = self.root["edificios"]
92             for id, edificio in edificios.items():
93                 logging.info(
94                     f"ID: {id}, Nombre: {edificio.nombre}, Ubicación: {edificio.ubicación}, "
95                     f"Año De Construcción: {edificio.año_construcción}, Estilo Arquitectónico: {edificio.estilo_arquitectónico}"
96                 )
97             return edificios
98         except Exception as e:
99             logging.error(f"Error al leer los edificios: {e}")
100    def actualizar_edificio(self, id, nombre, ubicacion, año_construcción, estilo_arquitectónico):
101        """Actualiza los atributos de un edificio."""
102        try:
103            edificio = self.root["edificios"].get(id)
104            if not edificio:
105                raise ValueError(f"No existe un edificio con ID {id}.")
106            edificio.nombre = nombre
107            edificio.ubicación = ubicacion
108            edificio.año_construcción = año_construcción
109            edificio.estilo_arquitectónico = estilo_arquitectónico
110            logging.info(f"Edificio con ID {id} actualizado exitosamente.")
111        except Exception as e:
112            logging.error(f"Error al actualizar el edificio con ID {id}: {e}")
113    def eliminar_edificio(self, id):
114        """Elimina un edificio por su ID."""
115        try:
116            if id not in self.root["edificios"]:
117                raise ValueError(f"No existe un edificio con ID {id}.")
118            del self.root["edificios"][id]
119            logging.info(f"Edificio con ID {id} eliminado exitosamente.")
120        except Exception as e:
121            logging.error(f"Error al eliminar el edificio con ID {id}: {e}")
122            raise ValueError(f"Ya existe un edificio con ID {id}.")
```

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** 5 de dic 10:28 actividad5.py - Act5 - Visual Studio Code
- File Explorer:** Shows a folder named "ACTS" containing files: venv_actividad5, actividad5.py, databasemanager_object.log, edificios.fs, edificios.fs.index, edificios.fs.lock, and edificios.fs.tmp.
- Code Editor:** The main window displays the Python script "actividad5.py". The code handles building operations like creation, reading, updating, and deletion, utilizing a DatabaseManagerObject.
- Status Bar:** Ln 153, Col 38 Spaces:4 UTF-8 LF {} Python 3.12.3 (venv_actividad5:venv)

The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** 5 de dic 10:28 actividad5.py - Act5 - Visual Studio Code
- File Explorer:** Shows a folder named "ACTS" containing files: venv_actividad5, actividad5.py, databasemanager_object.log, edificios.fs, edificios.fs.index, edificios.fs.lock, and edificios.fs.tmp.
- Terminal Tab:** The terminal tab displays the log output from running the script. It shows the creation of three buildings (Empire State, Louvre, and Torre Eiffel) and attempting to create a fourth building (Torre Eiffel again), which fails due to an existing entry.
- Status Bar:** Python 3.12.3 (venv_actividad5:venv)

```
2024-12-05 10:27:51,373 - INFO - Conexión establecida con ZODB.
2024-12-05 10:27:51,373 - INFO - Transacción iniciada.
2024-12-05 10:27:51,373 - INFO - Edificio con ID 1 creado exitosamente.
2024-12-05 10:27:51,373 - INFO - Edificio con ID 2 creado exitosamente.
2024-12-05 10:27:51,373 - INFO - Edificio con ID 3 creado exitosamente.
2024-12-05 10:27:51,373 - INFO - Transacción confirmada.
2024-12-05 10:27:51,373 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 12/11/1889, Estilo Arquitectónico: Arquitectura de Hierro
2024-12-05 10:27:51,373 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico: Art Deco
2024-12-05 10:27:51,373 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno
2024-12-05 10:27:51,373 - ERROR - Error al crear el edificio con ID 3: Ya existe un edificio con ID 3.
2024-12-05 10:27:51,373 - INFO - Transacción revertida.
2024-12-05 10:27:51,373 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 12/11/1889, Estilo Arquitectónico: Arquitectura de Hierro
2024-12-05 10:27:51,373 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico: Art Deco
2024-12-05 10:27:51,373 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno
2024-12-05 10:27:51,374 - INFO - Transacción iniciada.
2024-12-05 10:27:51,374 - INFO - Edificio con ID 1 actualizado exitosamente.
2024-12-05 10:27:51,374 - INFO - Transacción confirmada.
2024-12-05 10:27:51,374 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 11/12/1889, Estilo Arquitectónico: Arquitectura Moderna
2024-12-05 10:27:51,374 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico: Art Deco
2024-12-05 10:27:51,374 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno
2024-12-05 10:27:51,374 - INFO - Transacción iniciada.
2024-12-05 10:27:51,374 - ERROR - Error al eliminar el edificio con ID 7: No existe un edificio con ID 7.
2024-12-05 10:27:51,374 - INFO - Transacción revertida.
2024-12-05 10:27:51,374 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 11/12/1889, Estilo Arquitectónico: Arquitectura Moderna
2024-12-05 10:27:51,374 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico: Art Deco
2024-12-05 10:27:51,374 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno
2024-12-05 10:27:51,374 - INFO - Conexión a ZODB cerrada.
```

5 de dic 10:29

databasemanager_object.log - Act5 - Visual Studio Code

File Edit Selection View Go Run Terminal Help

actividad5.py databasemanager_object.log

```
1 2024-12-05 10:27:51,373 - INFO - Conexión establecida con ZODB.
2 2024-12-05 10:27:51,373 - INFO - Transacción iniciada.
3 2024-12-05 10:27:51,373 - INFO - Edificio con ID 1 creado exitosamente.
4 2024-12-05 10:27:51,373 - INFO - Edificio con ID 2 creado exitosamente.
5 2024-12-05 10:27:51,373 - INFO - Edificio con ID 3 creado exitosamente.
6 2024-12-05 10:27:51,373 - INFO - Transacción confirmada.
7 2024-12-05 10:27:51,373 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 12/11/1889, Estilo Arquitectónico.
8 2024-12-05 10:27:51,373 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico.
9 2024-12-05 10:27:51,373 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno.
10 2024-12-05 10:27:51,373 - INFO - Transacción iniciada.
11 2024-12-05 10:27:51,373 - ERROR - Error al crear el edificio con ID 3: Ya existe un edificio con ID 3.
12 2024-12-05 10:27:51,373 - INFO - Transacción revertida.
13 2024-12-05 10:27:51,373 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 12/11/1889, Estilo Arquitectónico.
14 2024-12-05 10:27:51,373 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico.
15 2024-12-05 10:27:51,373 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno.
16 2024-12-05 10:27:51,374 - INFO - Transacción iniciada.
17 2024-12-05 10:27:51,374 - INFO - Edificio con ID 1 actualizado exitosamente.
18 2024-12-05 10:27:51,374 - INFO - Transacción confirmada.
19 2024-12-05 10:27:51,374 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 11/12/1889, Estilo Arquitectónico.
20 2024-12-05 10:27:51,374 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico.
21 2024-12-05 10:27:51,374 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno.
22 2024-12-05 10:27:51,374 - INFO - Transacción iniciada.
23 2024-12-05 10:27:51,374 - ERROR - Error al eliminar el edificio con ID 7: No existe un edificio con ID 7.
24 2024-12-05 10:27:51,374 - INFO - Transacción revertida.
25 2024-12-05 10:27:51,374 - INFO - ID: 1, Nombre: Torre Eiffel, Ubicación: París, Año de Construcción: 11/12/1889, Estilo Arquitectónico.
26 2024-12-05 10:27:51,374 - INFO - ID: 2, Nombre: Empire State, Ubicación: Nueva York, Año de Construcción: 22/05/1931, Estilo Arquitectónico.
27 2024-12-05 10:27:51,374 - INFO - ID: 3, Nombre: Louvre, Ubicación: París, Año de Construcción: 01/06/1520, Estilo Arquitectónico: Moderno.
28 2024-12-05 10:27:51,374 - INFO - Conexión a ZODB cerrada.
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF Log