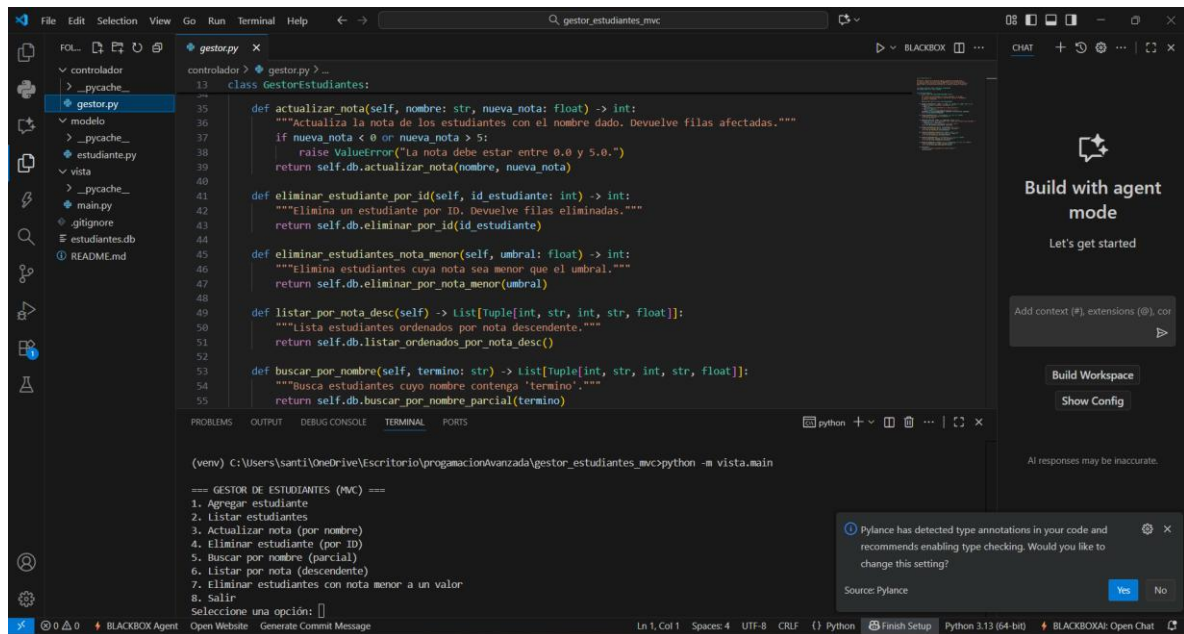


FUNCIONAMIENTO:

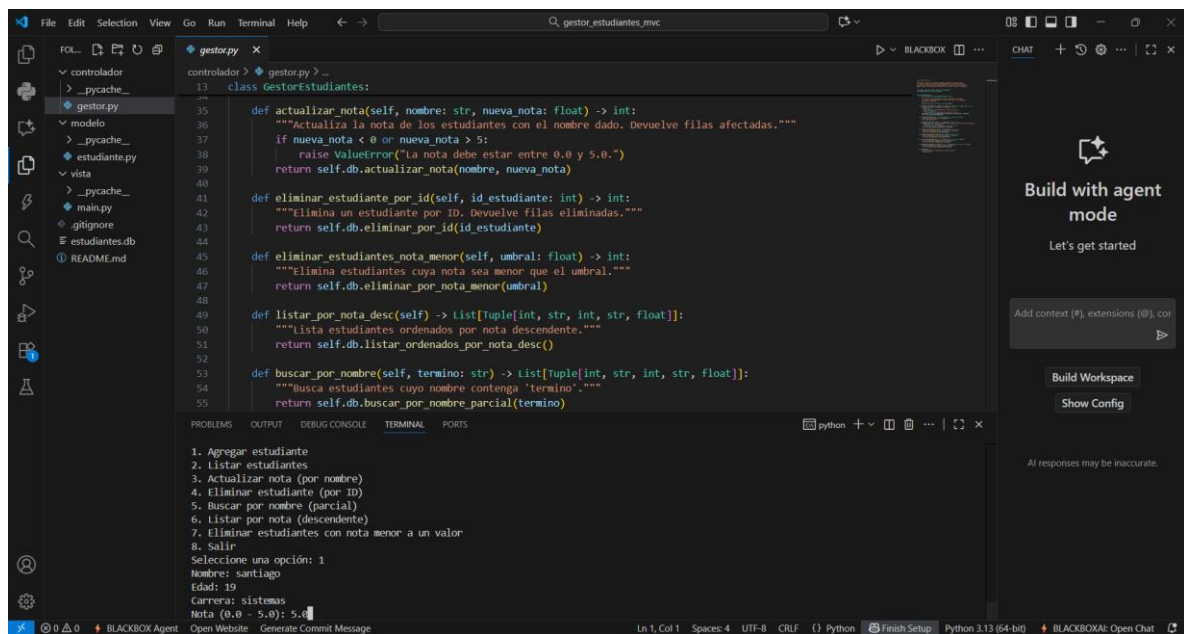


The screenshot shows a VS Code editor with the file explorer on the left displaying a project structure: `controlador`, `_pycache_`, `gestor.py`, `modelo`, `_pycache_`, `estudiante.py`, `vista`, `_pycache_`, `main.py`, `.gitignore`, `estudiantes.db`, and `README.md`. The `gestor.py` file is open in the editor, showing a `GestorEstudiantes` class with methods for updating notes, deleting students, and listing students. The terminal window at the bottom shows the command `(venv) C:\Users\santi\OneDrive\Escritorio\programacionAvanzada\gestor_estudiantes_mvc\python -m vista.main` and the output of the application, which lists the menu options.

```
controlador > gestor.py ...
13 class GestorEstudiantes:
...
35 def actualizar_nota(self, nombre: str, nueva_nota: float) -> int:
36     """Actualiza la nota de los estudiantes con el nombre dado. Devuelve filas afectadas."""
37     if nueva_nota < 0 or nueva_nota > 5:
38         raise ValueError("La nota debe estar entre 0.0 y 5.0.")
39     return self.db.actualizar_nota(nombre, nueva_nota)
40
41 def eliminar_estudiante_por_id(self, id_estudiante: int) -> int:
42     """Elimina un estudiante por ID. Devuelve filas eliminadas."""
43     return self.db.eliminar_por_id(id_estudiante)
44
45 def eliminar_estudiantes_nota_menor(self, umbral: float) -> int:
46     """Elimina estudiantes cuya nota sea menor que el umbral."""
47     return self.db.eliminar_por_nota_menor(umbral)
48
49 def listar_por_nota_desc(self) -> List[Tuple[int, str, int, str, float]]:
50     """Lista estudiantes ordenados por nota descendente."""
51     return self.db.listar_ordenados_por_nota_desc()
52
53 def buscar_por_nombre(self, termino: str) -> List[Tuple[int, str, int, str, float]]:
54     """Busca estudiantes cuyo nombre contenga 'termino'."""
55     return self.db.buscar_por_nombre_parcial(termino)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(venv) C:\Users\santi\OneDrive\Escritorio\programacionAvanzada\gestor_estudiantes_mvc\python -m vista.main

=== GESTOR DE ESTUDIANTES (MVC) ===
1. Agregar estudiante
2. Listar estudiantes
3. Actualizar nota (por nombre)
4. Eliminar estudiante (por ID)
5. Buscar por nombre (parcial)
6. Listar por nota (descendente)
7. Eliminar estudiantes con nota menor a un valor
8. Salir
Seleccione una opción: []
```



This screenshot shows the same VS Code editor setup as the previous one, but the terminal window now displays the user's input for the application. The user has selected option 1, entered the name 'Santiago', the ID '10', and the note '5.0'.

```
1. Agregar estudiante
2. Listar estudiantes
3. Actualizar nota (por nombre)
4. Eliminar estudiante (por ID)
5. Buscar por nombre (parcial)
6. Listar por nota (descendente)
7. Eliminar estudiantes con nota menor a un valor
8. Salir
Seleccione una opción: 1
Nombre: Santiago
Idado: 10
Carrera: sistemas
Nota (0.0 - 5.0): 5.0
```

File Edit Selection View Go Run Terminal Help

gestor_estudiantes_mvc

CHAT + [Icons]

gestor.py

controlador > gestor.py > ...

class GestorEstudiantes:

13

def actualizar_nota(self, nombre: str, nueva_nota: float) -> int:

35

"""Actualiza la nota de los estudiantes con el nombre dado. Devuelve filas afectadas."""

36

if nueva_nota < 0 or nueva_nota > 5:

37

raise ValueError("La nota debe estar entre 0.0 y 5.0.")

38

return self.db.actualizar_nota(nombre, nueva_nota)

39

40

def eliminar_estudiante_por_id(self, id_estudiante: int) -> int:

41

"""Elimina un estudiante por ID. Devuelve filas eliminadas."""

42

return self.db.eliminar_por_id(id_estudiante)

43

44

def eliminar_estudiantes_nota_menor(self, umbral: float) -> int:

45

"""Elimina estudiantes cuya nota sea menor que el umbral."""

46

return self.db.eliminar_por_nota_menor(umbral)

47

48

def listar_por_nota_desc(self) -> List[Tuple[int, str, int, str, float]]:

49

"""Lista estudiantes ordenados por nota descendente."""

50

return self.db.listar_ordenados_por_nota_desc()

51

52

def buscar_por_nombre(self, termino: str) -> List[Tuple[int, str, int, str, float]]:

53

"""Busca estudiantes cuyo nombre contenga 'termino'."""

54

return self.db.buscar_por_nombre_parcial(termino)

55

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

python + [Icons]

8. Salir

Seleccione una opción: 2

ID Nombre Edad Carrera Nota

1 santiago 19 sistemas 5.0

2 santiago 19 sistemas 5.0

===== GESTOR DE ESTUDIANTES (PMC) =====

1. Agregar estudiante

2. Listar estudiantes

3. Actualizar nota (por nombre)

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

Finish Setup Python 3.13 (64-bit) BLACKBOXAI: Open Chat

File Edit Selection View Go Run Terminal Help

gestor_estudiantes_mvc

CHAT + [Icons]

gestor.py

controlador > gestor.py > ...

class GestorEstudiantes:

13

def actualizar_nota(self, nombre: str, nueva_nota: float) -> int:

35

"""Actualiza la nota de los estudiantes con el nombre dado. Devuelve filas afectadas."""

36

if nueva_nota < 0 or nueva_nota > 5:

37

raise ValueError("La nota debe estar entre 0.0 y 5.0.")

38

return self.db.actualizar_nota(nombre, nueva_nota)

39

40

def eliminar_estudiante_por_id(self, id_estudiante: int) -> int:

41

"""Elimina un estudiante por ID. Devuelve filas eliminadas."""

42

return self.db.eliminar_por_id(id_estudiante)

43

44

def eliminar_estudiantes_nota_menor(self, umbral: float) -> int:

45

"""Elimina estudiantes cuya nota sea menor que el umbral."""

46

return self.db.eliminar_por_nota_menor(umbral)

47

48

def listar_por_nota_desc(self) -> List[Tuple[int, str, int, str, float]]:

49

"""Lista estudiantes ordenados por nota descendente."""

50

return self.db.listar_ordenados_por_nota_desc()

51

52

def buscar_por_nombre(self, termino: str) -> List[Tuple[int, str, int, str, float]]:

53

"""Busca estudiantes cuyo nombre contenga 'termino'."""

54

return self.db.buscar_por_nombre_parcial(termino)

55

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

python + [Icons]

7. Eliminar estudiantes con nota menor a un valor

8. Salir

Seleccione una opción: 2

ID Nombre Edad Carrera Nota

1 santiago 19 sistemas 5.0

2 santiago 19 sistemas 5.0

3 ana 19 comunicacion 5.0

===== GESTOR DE ESTUDIANTES (PMC) =====

1. Agregar estudiante

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

Finish Setup Python 3.13 (64-bit) BLACKBOXAI: Open Chat

```
13 class GestorEstudiantes:
14
15     def actualizar_nota(self, nombre: str, nueva_nota: float) -> int:
16         """Actualiza la nota de los estudiantes con el nombre dado. Devuelve filas afectadas."""
17         if nueva_nota < 0 or nueva_nota > 5:
18             raise ValueError("La nota debe estar entre 0.0 y 5.0.")
19         return self.db.actualizar_nota(nombre, nueva_nota)
20
21     def eliminar_estudiante_por_id(self, id_estudiante: int) -> int:
22         """Elimina un estudiante por ID. Devuelve filas eliminadas."""
23         return self.db.eliminar_por_id(id_estudiante)
24
25     def eliminar_estudiantes_nota_menor(self, umbral: float) -> int:
26         """Elimina estudiantes cuya nota sea menor que el umbral."""
27         return self.db.eliminar_por_nota_menor(umbral)
28
29     def listar_por_nota_desc(self) -> List[tuple[int, str, int, str, float]]:
30         """Lista estudiantes ordenados por nota descendente."""
31         return self.db.listar_ordenados_por_nota_desc()
32
33     def buscar_por_nombre(self, termino: str) -> List[tuple[int, str, int, str, float]]:
34         """Busca estudiantes cuyo nombre contenga 'termino'."""
35         return self.db.buscar_por_nombre parcial(termino)
```

8. Salir
Seleccione una opción: 2

ID	Nombre	Edad	Carrera	Nota
1	santiago	19	sistemas	5.0
2	santiago	19	sistemas	5.0
3	ana	19	comunicacion	5.0
4	Juan	20	electronica	2.0
5	pedro	21	sistemas	3.2

=== GESTOR DE ESTUDIANTES (MVC) ===

```
6. Listar por nota (descendente)
7. Eliminar estudiantes con nota menor a un valor
8. Salir
Seleccione una opción: 5
Cadena a buscar en nombres: ana
```

ID	Nombre	Edad	Carrera	Nota
3	ana	19	comunicacion	5.0

=== GESTOR DE ESTUDIANTES (MVC) ===

1. Agregar estudiante

```
7. Eliminar estudiantes con nota menor a un valor
8. Salir
Seleccione una opción: 7
Eliminar estudiantes con nota menor que: 4.3
❌ Debe ser <= 5.0
Eliminar estudiantes con nota menor que: 4.0
✅ Se eliminaron 2 estudiante(s).
```

=== GESTOR DE ESTUDIANTES (MVC) ===

1. Agregar estudiante
2. Listar estudiantes
3. Actualizar nota (por nombre)
4. Eliminar estudiante (por ID)

PREGUNTAS:

¿Qué ventajas encontraste al aplicar el patrón MVC en tu proyecto?

El patron MVC en mi proyecto me ayuda a tener las cosas mucho mas ordenadas y además que lo puedo entender mas fácil también. Ayuda al mantenimiento y cambiar partes de código sin afectar otras ya que esta bien ordenado.

¿Qué dificultades surgieron al separar las responsabilidades entre modelo, vista y controlador?

Lo difícil es organizar que código iba en cada lado, mas que nada entre el modelo y la vista

¿Qué aprendiste sobre la ejecución de sentencias SQL y su relación con la estructura del código?

De las sentencias aprendí que deben ser desde el modelo, entendí que es mejor usar parámetros para prevenir las inyecciones

¿Qué mejorarías si tuvieras que ampliar el sistema?

Siento que estaría bueno mejorar la interfaz y también se podría ampliar el uso si se pueden escoger mas roles y que no sea solo para estudiantes