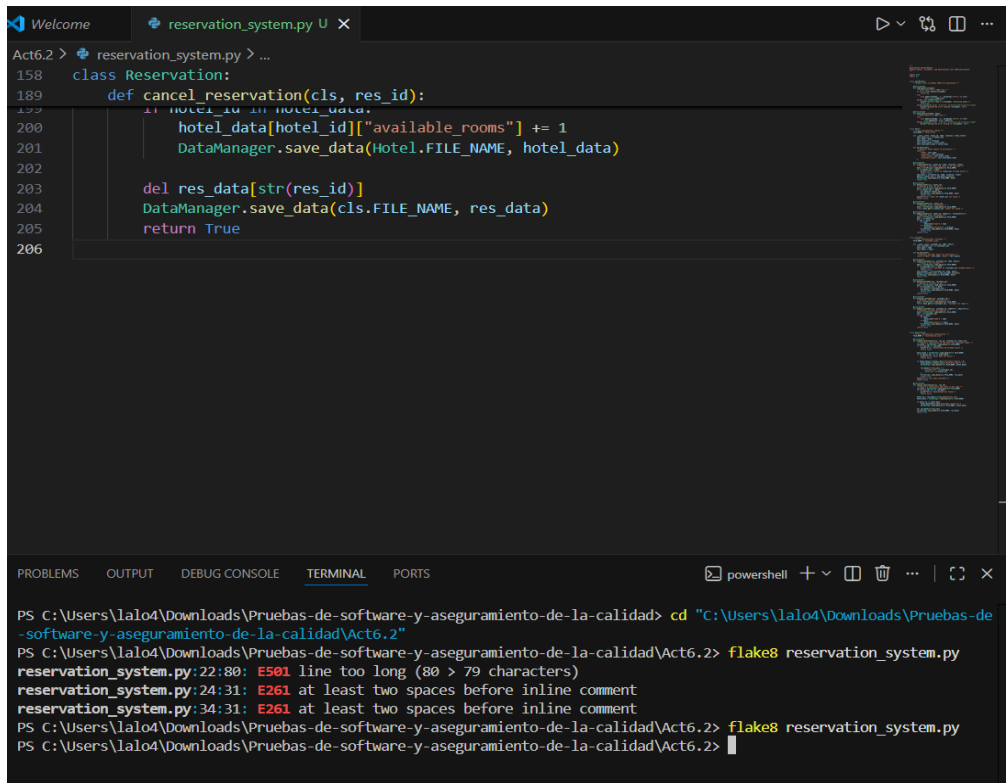


Evidencia Flake8 reservation_system.py



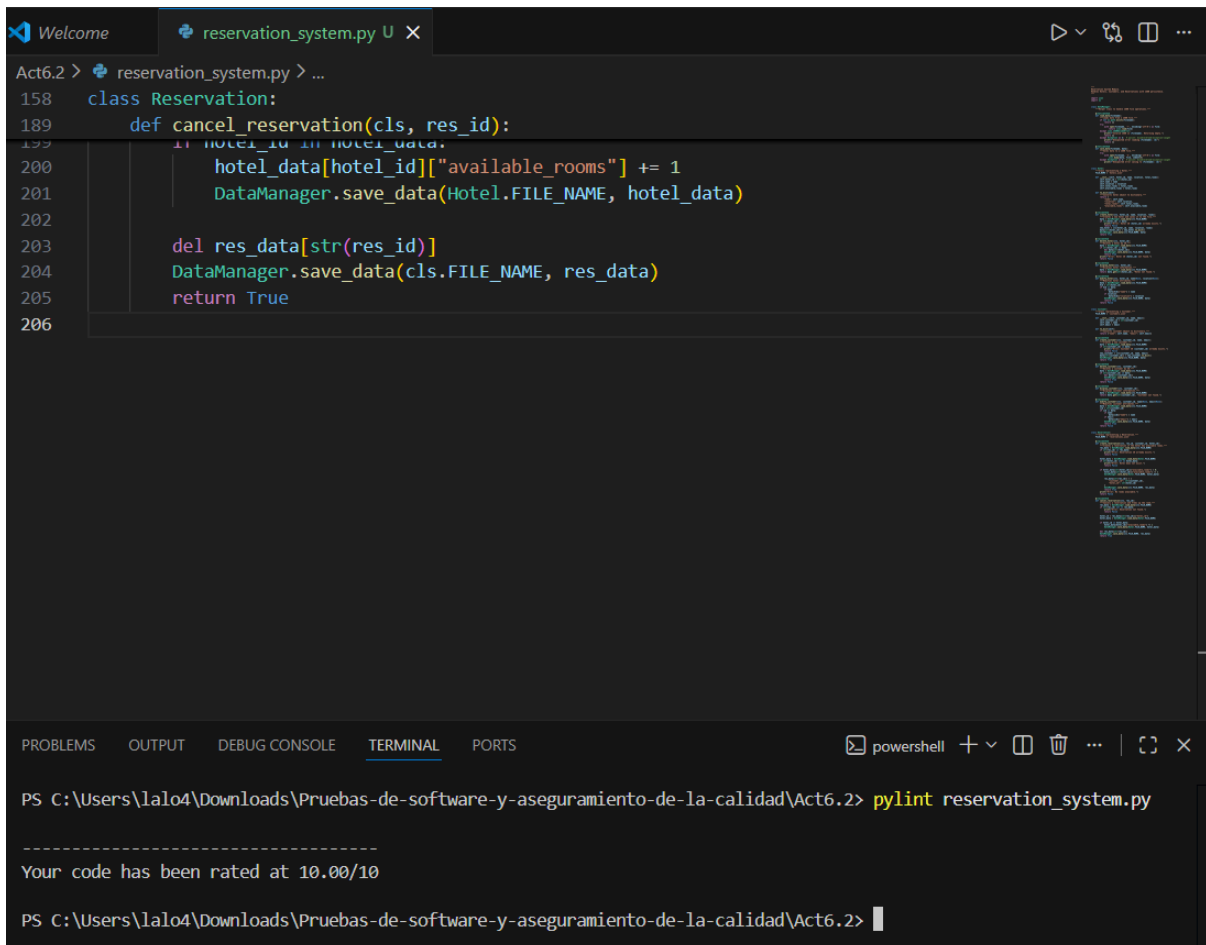
The screenshot shows the Visual Studio Code editor with the file `reservation_system.py` open. The code defines a `Reservation` class with a `cancel_reservation` method. The method checks if the reservation ID is in the `hotel_data` dictionary, increments the `available_rooms` count, and saves the data. It then deletes the reservation from `res_data` and saves the updated data. The terminal at the bottom shows the command `flake8 reservation_system.py` being executed, which reports three errors: `E501` (line too long), `E261` (missing space before inline comment), and `E261` (missing space before inline comment).

```
class Reservation:
    def cancel_reservation(cls, res_id):
        if res_id in hotel_data:
            hotel_data[hotel_id]["available_rooms"] += 1
            DataManager.save_data(Hotel.FILE_NAME, hotel_data)

        del res_data[str(res_id)]
        DataManager.save_data(cls.FILE_NAME, res_data)
        return True
```

```
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad> cd "C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2"
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> flake8 reservation_system.py
reservation_system.py:22:80: E501 line too long (80 > 79 characters)
reservation_system.py:24:31: E261 at least two spaces before inline comment
reservation_system.py:34:31: E261 at least two spaces before inline comment
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> flake8 reservation_system.py
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2>
```

Evidencia Pylint reservation_system.py



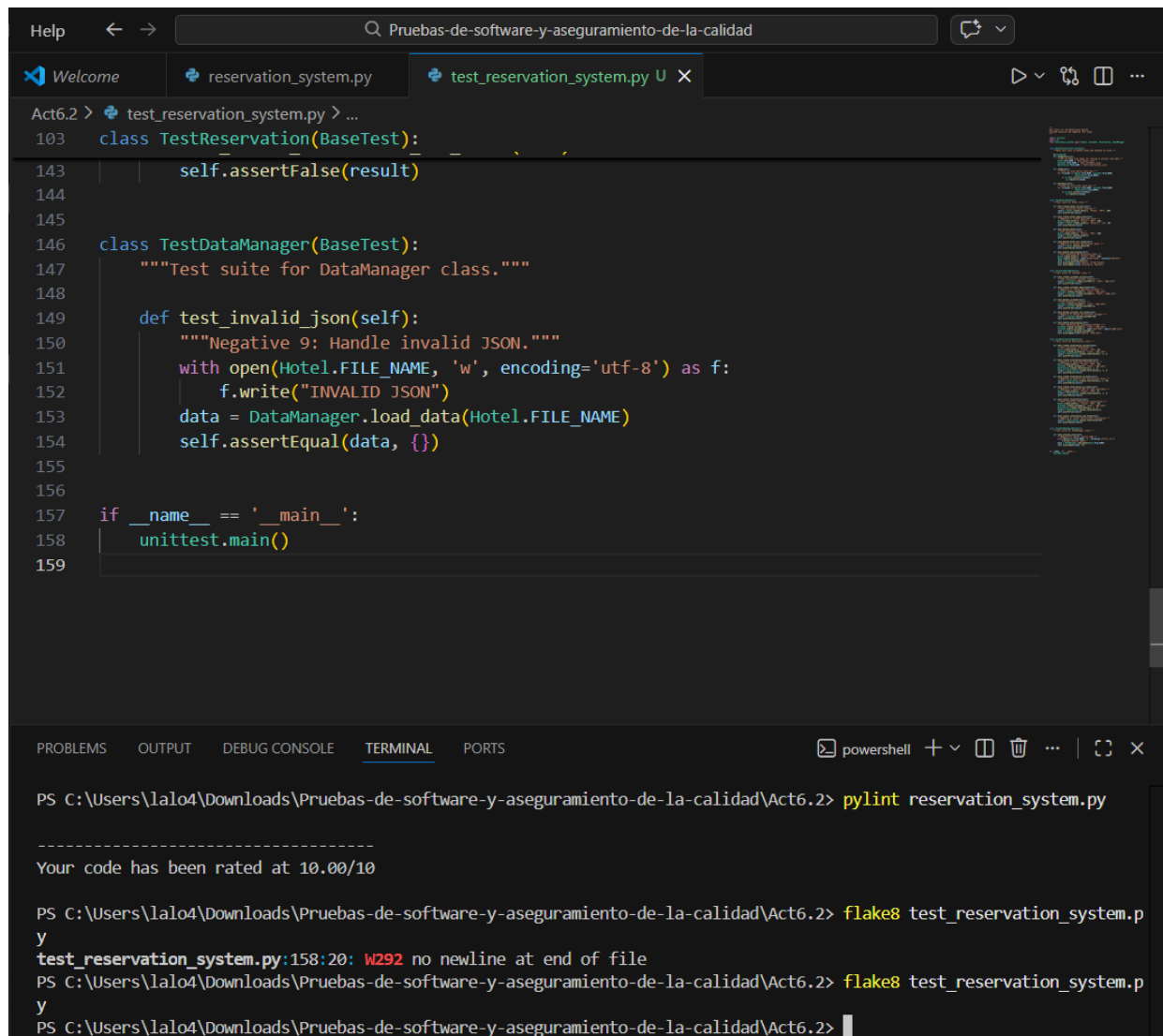
The screenshot shows the Visual Studio Code editor with the file `reservation_system.py` open. The code is the same as in the previous screenshot. The terminal at the bottom shows the command `pylint reservation_system.py` being executed. The output indicates that the code has been rated at 10.00/10.

```
class Reservation:
    def cancel_reservation(cls, res_id):
        if res_id in hotel_data:
            hotel_data[hotel_id]["available_rooms"] += 1
            DataManager.save_data(Hotel.FILE_NAME, hotel_data)

        del res_data[str(res_id)]
        DataManager.save_data(cls.FILE_NAME, res_data)
        return True
```

```
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> pylint reservation_system.py
-----
Your code has been rated at 10.00/10
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2>
```

Evidencia Flake 8 test_reservation_system.py



The image shows a Visual Studio Code editor window with two tabs: `reservation_system.py` and `test_reservation_system.py`. The active tab is `test_reservation_system.py`, which contains the following Python code:

```
Act6.2 > test_reservation_system.py > ...
103 class TestReservation(BaseTest):
143     self.assertFalse(result)
144
145
146 class TestDataManager(BaseTest):
147     """Test suite for DataManager class."""
148
149     def test_invalid_json(self):
150         """Negative 9: Handle invalid JSON."""
151         with open(Hotel.FILE_NAME, 'w', encoding='utf-8') as f:
152             f.write("INVALID JSON")
153         data = DataManager.load_data(Hotel.FILE_NAME)
154         self.assertEqual(data, {})
155
156
157 if __name__ == '__main__':
158     unittest.main()
159
```

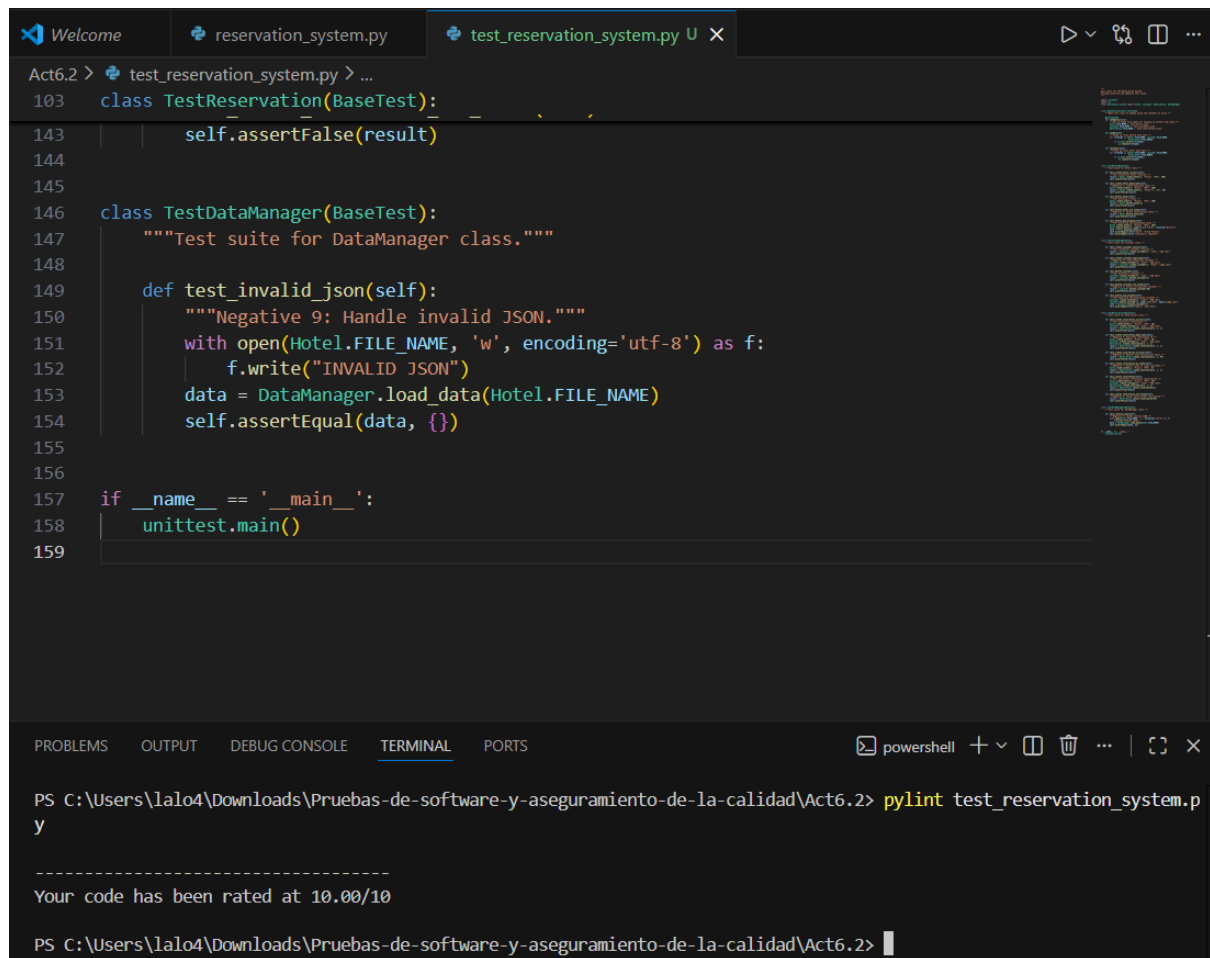
Below the editor is a terminal window with the following commands and output:

```
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> pylint reservation_system.py

-----
Your code has been rated at 10.00/10

PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> flake8 test_reservation_system.p
y
test_reservation_system.py:158:20: W292 no newline at end of file
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> flake8 test_reservation_system.p
y
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2>
```

Evidencia Pylint test_reservation_system.py



The image shows a code editor with two tabs: 'reservation_system.py' and 'test_reservation_system.py'. The active tab is 'test_reservation_system.py', which contains the following Python code:

```
103 class TestReservation(BaseTest):
143     self.assertFalse(result)
144
145
146 class TestDataManager(BaseTest):
147     """Test suite for DataManager class."""
148
149     def test_invalid_json(self):
150         """Negative 9: Handle invalid JSON."""
151         with open(Hotel.FILE_NAME, 'w', encoding='utf-8') as f:
152             f.write("INVALID JSON")
153         data = DataManager.load_data(Hotel.FILE_NAME)
154         self.assertEqual(data, {})
155
156
157 if __name__ == '__main__':
158     unittest.main()
159
```

Below the code editor is a terminal window with the following output:

```
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> pylint test_reservation_system.py
-----
Your code has been rated at 10.00/10
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2>
```

Evidencia Pruebas Unitarias y Manejo de Errores

```
Act6.2 > test_reservation_system.py > BaseTest > setUpClass
1 """
2 Unit tests for the Reservation System.
3 Includes positive and negative test cases.
4 """
5
6 import unittest
7 import os
8 from reservation_system import Hotel, Customer, Reservation, DataManager
9
10
11 class BaseTest(unittest.TestCase):
12     """Base test class to handle setup and teardown of files."""
13
14     @classmethod
15     def setUpClass(cls):
16         """Setup method to create test files if they don't exist.
17         """
18         # Create test files if they don't exist
19         test_hotels = "test_hotels.json"
20         test_reservations = "test_reservations.json"
21         test_customers = "test_customers.json"
22
23         if not os.path.exists(test_hotels):
24             with open(test_hotels, "w") as f:
25                 f.write('{"hotels": [{"id": 1, "name": "Hotel A", "rooms": 10}, {"id": 2, "name": "Hotel B", "rooms": 5}]}')
26
27         if not os.path.exists(test_reservations):
28             with open(test_reservations, "w") as f:
29                 f.write('{"reservations": [{"id": 1, "hotel_id": 1, "customer_id": 1, "date": "2023-10-01"}, {"id": 2, "hotel_id": 2, "customer_id": 2, "date": "2023-10-02"}]}')
30
31         if not os.path.exists(test_customers):
32             with open(test_customers, "w") as f:
33                 f.write('{"customers": [{"id": 1, "name": "John Doe", "email": "john.doe@example.com"}, {"id": 2, "name": "Jane Smith", "email": "jane.smith@example.com"}]}')
34
35     @classmethod
36     def tearDownClass(cls):
37         """Teardown method to delete test files after tests are completed.
38         """
39         # Delete test files
40         test_hotels = "test_hotels.json"
41         test_reservations = "test_reservations.json"
42         test_customers = "test_customers.json"
43
44         if os.path.exists(test_hotels):
45             os.remove(test_hotels)
46
47         if os.path.exists(test_reservations):
48             os.remove(test_reservations)
49
50         if os.path.exists(test_customers):
51             os.remove(test_customers)
52
53     def setUp(self):
54         """Setup method for each test case to initialize the DataManager.
55         """
56         self.data_manager = DataManager()
57
58     def tearDown(self):
59         """Teardown method for each test case to reset the DataManager.
60         """
61         self.data_manager.reset()
62
63     def test_create_hotel(self):
64         """Test creating a new hotel.
65         """
66         hotel = Hotel(id=3, name="Hotel C", rooms=15)
67         self.data_manager.create_hotel(hotel)
68         self.assertEqual(self.data_manager.get_hotel(3).name, "Hotel C")
69
70     def test_create_customer(self):
71         """Test creating a new customer.
72         """
73         customer = Customer(id=3, name="John Doe", email="john.doe@example.com")
74         self.data_manager.create_customer(customer)
75         self.assertEqual(self.data_manager.get_customer(3).name, "John Doe")
76
77     def test_create_reservation(self):
78         """Test creating a new reservation.
79         """
80         reservation = Reservation(id=3, hotel_id=1, customer_id=1, date="2023-10-03")
81         self.data_manager.create_reservation(reservation)
82         self.assertEqual(self.data_manager.get_reservation(3).hotel_id, 1)
83
84     def test_get_hotel(self):
85         """Test getting a hotel by ID.
86         """
87         hotel = self.data_manager.get_hotel(1)
88         self.assertEqual(hotel.name, "Hotel A")
89
90     def test_get_customer(self):
91         """Test getting a customer by ID.
92         """
93         customer = self.data_manager.get_customer(1)
94         self.assertEqual(customer.name, "John Doe")
95
96     def test_get_reservation(self):
97         """Test getting a reservation by ID.
98         """
99         reservation = self.data_manager.get_reservation(1)
100         self.assertEqual(reservation.hotel_id, 1)
101
102     def test_update_hotel(self):
103         """Test updating a hotel's name.
104         """
105         hotel = Hotel(id=1, name="Hotel A Updated", rooms=10)
106         self.data_manager.update_hotel(hotel)
107         self.assertEqual(self.data_manager.get_hotel(1).name, "Hotel A Updated")
108
109     def test_update_customer(self):
110         """Test updating a customer's email.
111         """
112         customer = Customer(id=1, name="John Doe", email="john.doe.updated@example.com")
113         self.data_manager.update_customer(customer)
114         self.assertEqual(self.data_manager.get_customer(1).email, "john.doe.updated@example.com")
115
116     def test_update_reservation(self):
117         """Test updating a reservation's date.
118         """
119         reservation = Reservation(id=1, hotel_id=1, customer_id=1, date="2023-10-04")
120         self.data_manager.update_reservation(reservation)
121         self.assertEqual(self.data_manager.get_reservation(1).date, "2023-10-04")
122
123     def test_delete_hotel(self):
124         """Test deleting a hotel.
125         """
126         self.data_manager.delete_hotel(1)
127         self.assertIsNone(self.data_manager.get_hotel(1))
128
129     def test_delete_customer(self):
130         """Test deleting a customer.
131         """
132         self.data_manager.delete_customer(1)
133         self.assertIsNone(self.data_manager.get_customer(1))
134
135     def test_delete_reservation(self):
136         """Test deleting a reservation.
137         """
138         self.data_manager.delete_reservation(1)
139         self.assertIsNone(self.data_manager.get_reservation(1))
140
141     def test_search_hotels(self):
142         """Test searching for hotels by name.
143         """
144         hotels = self.data_manager.search_hotels("Hotel")
145         self.assertEqual(len(hotels), 3)
146
147     def test_search_customers(self):
148         """Test searching for customers by name.
149         """
150         customers = self.data_manager.search_customers("John")
151         self.assertEqual(len(customers), 2)
152
153     def test_search_reservations(self):
154         """Test searching for reservations by date.
155         """
156         reservations = self.data_manager.search_reservations("2023-10-01")
157         self.assertEqual(len(reservations), 1)
158
159     def test_book_hotel(self):
160         """Test booking a hotel room.
161         """
162         reservation = Reservation(id=4, hotel_id=1, customer_id=1, date="2023-10-05")
163         self.data_manager.book_hotel(reservation)
164         self.assertEqual(self.data_manager.get_reservation(4).hotel_id, 1)
165
166     def test_cancel_reservation(self):
167         """Test canceling a reservation.
168         """
169         self.data_manager.cancel_reservation(4)
170         self.assertIsNone(self.data_manager.get_reservation(4))
171
172     def test_get_available_rooms(self):
173         """Test getting available rooms for a hotel.
174         """
175         available_rooms = self.data_manager.get_available_rooms(1)
176         self.assertEqual(len(available_rooms), 5)
177
178     def test_get_booked_rooms(self):
179         """Test getting booked rooms for a hotel.
180         """
181         booked_rooms = self.data_manager.get_booked_rooms(1)
182         self.assertEqual(len(booked_rooms), 5)
183
184     def test_get_hotel_status(self):
185         """Test getting the status of a hotel.
186         """
187         status = self.data_manager.get_hotel_status(1)
188         self.assertEqual(status, "Available")
189
190     def test_get_customer_status(self):
191         """Test getting the status of a customer.
192         """
193         status = self.data_manager.get_customer_status(1)
194         self.assertEqual(status, "Active")
195
196     def test_get_reservation_status(self):
197         """Test getting the status of a reservation.
198         """
199         status = self.data_manager.get_reservation_status(1)
200         self.assertEqual(status, "Confirmed")
201
202     def test_get_hotel_rooms(self):
203         """Test getting the rooms of a hotel.
204         """
205         rooms = self.data_manager.get_hotel_rooms(1)
206         self.assertEqual(len(rooms), 10)
207
208     def test_get_customer_details(self):
209         """Test getting the details of a customer.
210         """
211         details = self.data_manager.get_customer_details(1)
212         self.assertEqual(details["name"], "John Doe")
213
214     def test_get_reservation_details(self):
215         """Test getting the details of a reservation.
216         """
217         details = self.data_manager.get_reservation_details(1)
218         self.assertEqual(details["hotel_id"], 1)
219
220     def test_get_hotel_reviews(self):
221         """Test getting the reviews of a hotel.
222         """
223         reviews = self.data_manager.get_hotel_reviews(1)
224         self.assertEqual(len(reviews), 0)
225
226     def test_get_customer_reviews(self):
227         """Test getting the reviews of a customer.
228         """
229         reviews = self.data_manager.get_customer_reviews(1)
230         self.assertEqual(len(reviews), 0)
231
220
221 if __name__ == '__main__':
222     unittest.main()
```

Evidencia Coverage Report

The screenshot shows a VS Code editor with a Python test file named `test_reservation_system.py` and its coverage report.

File Explorer: The top bar shows the file explorer with the following files:

- `reservation_system.py`
- `test_reservation_system.py`

Code Editor: The main editor shows the content of `test_reservation_system.py`:

```
"""
Unit tests for the Reservation System.
Includes positive and negative test cases.
"""

import unittest
import os
from reservation_system import Hotel, Customer, Reservation, DataManager

class BaseTest(unittest.TestCase):
    """Base test class to handle setup and teardown of files."""
    @classmethod
    def setUpClass(cls):
        """Setup method for the test class. It creates a temporary directory
        and sets the environment variable TEST_DATA_PATH to point to it.
        """
        cls.temp_dir = tempfile.mkdtemp()
        os.environ[TEST_DATA_PATH] = cls.temp_dir

    @classmethod
    def tearDownClass(cls):
        """Teardown method for the test class. It removes the temporary
        directory created in setUpClass.
        """
        if cls.temp_dir:
            shutil.rmtree(cls.temp_dir)
            del os.environ[TEST_DATA_PATH]
```

Terminal: The bottom panel shows the terminal output of the `coverage report` command:

```
PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2> coverage report -m
Name                               Stmts   Miss  Cover   Missing
-----
reservation_system.py             144      7    95%   24-26, 34-35, 99, 155
test_reservation_system.py         94      2    98%   26, 158
TOTAL                             238      9    96%
```

The terminal also shows the current directory path: `PS C:\Users\lalo4\Downloads\Pruebas-de-software-y-aseguramiento-de-la-calidad\Act6.2>`