

Pruebas de software y aseguramiento de la calidad

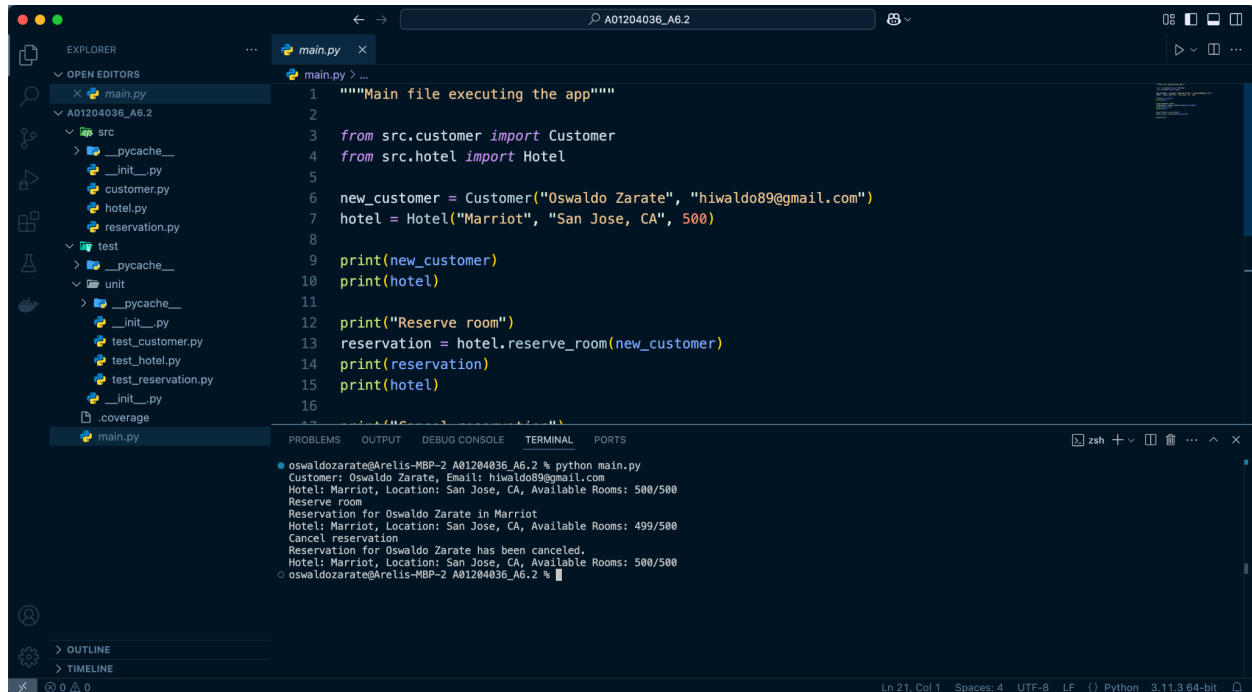
Tarea. Pruebas de unidad

Oswaldo Zárate Cisneros A01204036

16 de febrero del 2025

Pruebas de ejecución

Dentro del archivo main.py creé un ejemplo de ejecución de la aplicación. El archivo incluye la creación de una instancia de la clase Customer y Hotel. Se ejecuta el método reserve_room de la clase Hotel y el método cancel_reservation de la misma clase. La clase Reservation es usada por la clase Hotel.

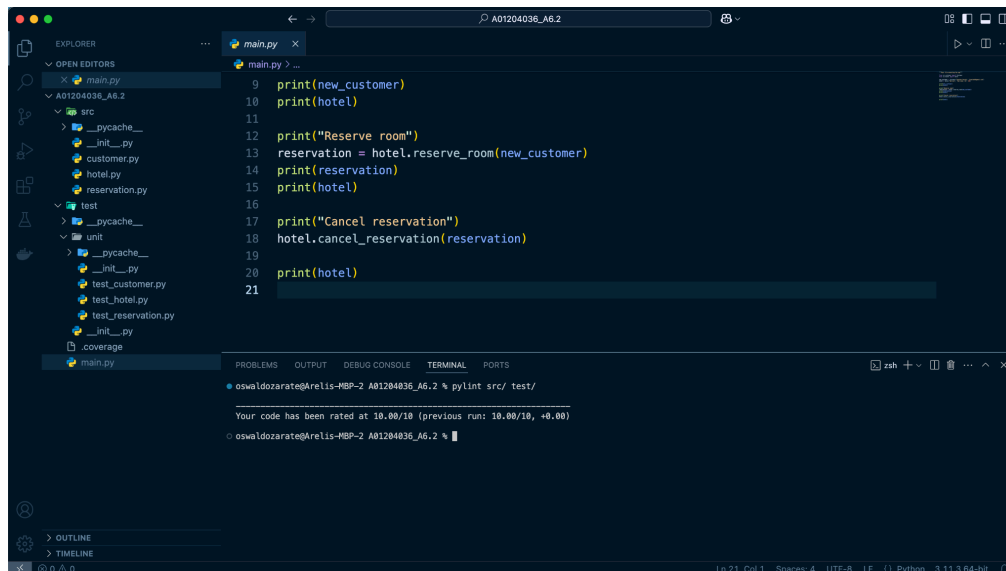


```
1 """Main file executing the app"""
2
3 from src.customer import Customer
4 from src.hotel import Hotel
5
6 new_customer = Customer("Oswaldo Zarate", "hiwaldo89@gmail.com")
7 hotel = Hotel("Marriot", "San Jose, CA", 500)
8
9 print(new_customer)
10 print(hotel)
11
12 print("Reserve room")
13 reservation = hotel.reserve_room(new_customer)
14 print(reservation)
15 print(hotel)
16
```

```
ooswaldozarate@Arelis-MBP-2 A01204036_A6.2 % python main.py
Customer: Oswaldo Zarate, Email: hiwaldo89@gmail.com
Hotel: Marriot, Location: San Jose, CA, Available Rooms: 500/500
Reserve room
Reservation for Oswaldo Zarate in Marriot
Hotel: Marriot, Location: San Jose, CA, Available Rooms: 499/500
Cancel reservation
Reservation for Oswaldo Zarate has been canceled.
Hotel: Marriot, Location: San Jose, CA, Available Rooms: 500/500
ooswaldozarate@Arelis-MBP-2 A01204036_A6.2 %
```

PEP8

Utilicé pylint para verificar que se sigan las convenciones de PEP8. A continuación se muestra el resultado de la ejecución. Se verifican los archivos dentro de los directorios src y test.

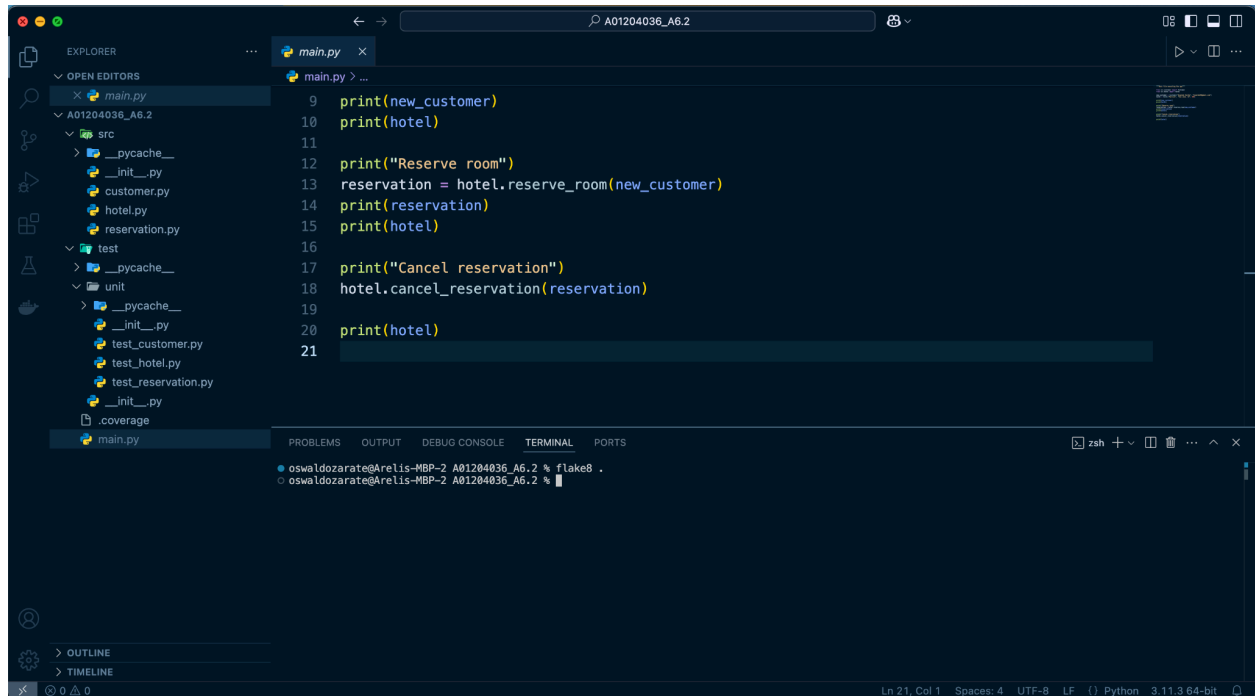


```
9 print(new_customer)
10 print(hotel)
11
12 print("Reserve room")
13 reservation = hotel.reserve_room(new_customer)
14 print(reservation)
15 print(hotel)
16
17 print("Cancel reservation")
18 hotel.cancel_reservation(reservation)
19
20 print(hotel)
21
```

```
ooswaldozarate@Arelis-MBP-2 A01204036_A6.2 % pylint src/ test/
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
ooswaldozarate@Arelis-MBP-2 A01204036_A6.2 %
```

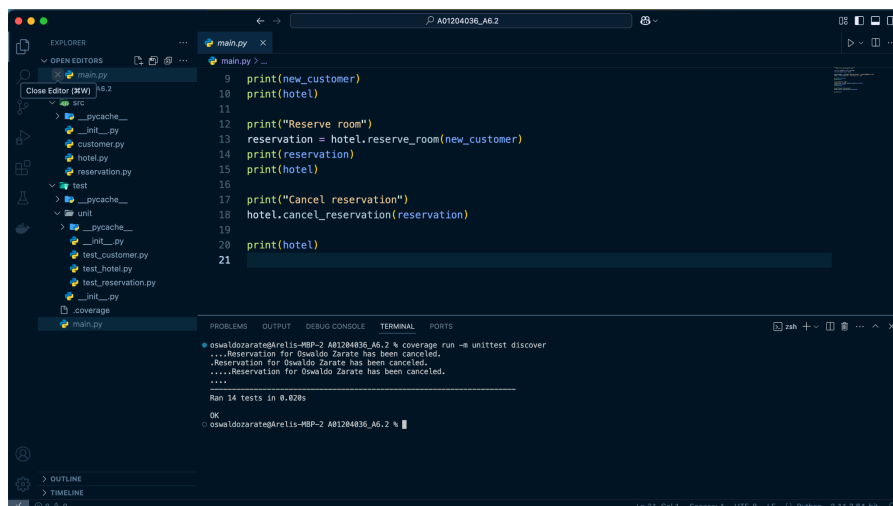
Flake8

Utilicé Flake8 para verificar el estilo del código. Podemos ver en la siguiente imagen el resultado de ejecutar Flake8 en todos los archivos py. No se muestra ningún error, lo que nos dice que el estilo del código es correcto.

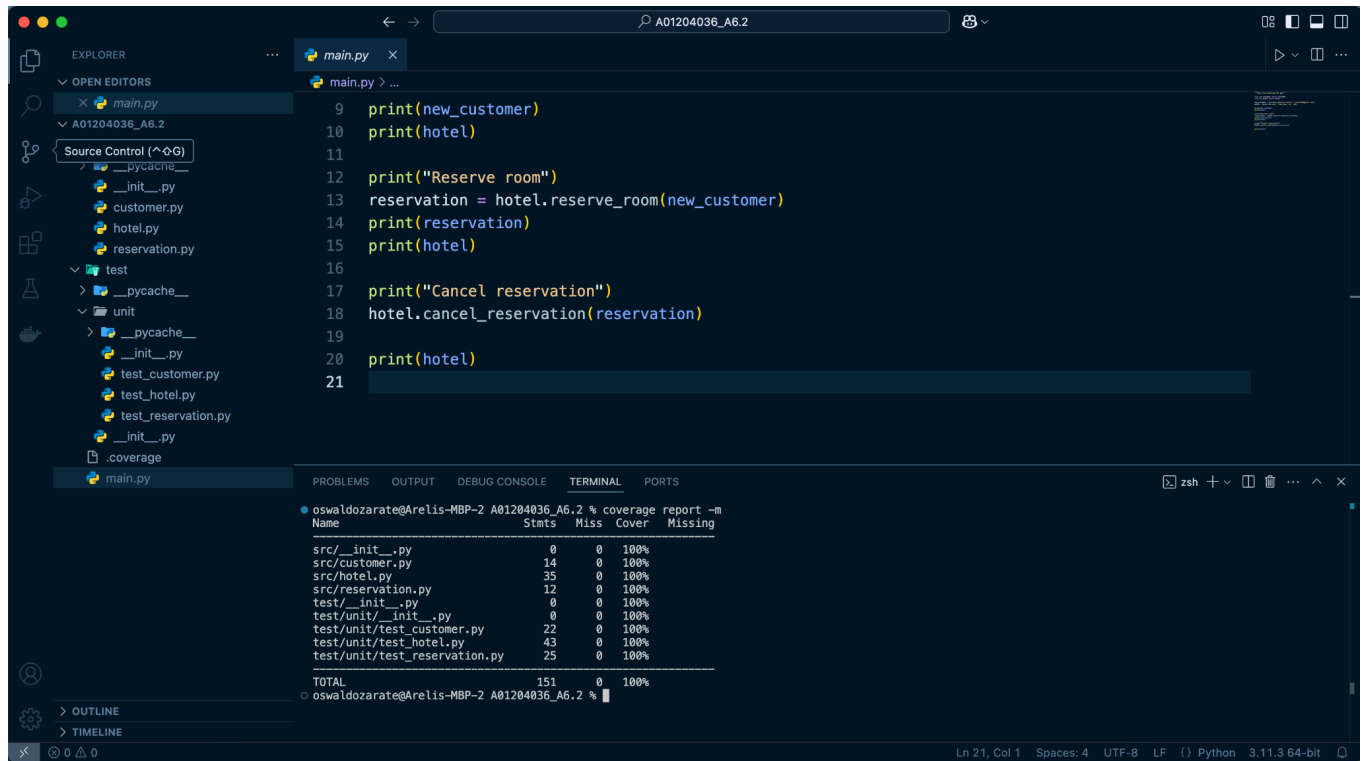


Pruebas unitarias

Para realizar pruebas unitarias, creé archivos de pruebas para cada clase dentro del directorio test. Utilicé las librerías unittest y coverage para correr las pruebas. Primero ejecuté `coverage run -m unittest discover` para ejecutar todas las pruebas. Podemos ver que se ejecutaron 14 pruebas y todas pasaron.



Después de eso ejecuté *coverage report -m* para generar el reporte de cobertura. Podemos ver que todas las líneas de código están cubiertas.



The screenshot shows a Visual Studio Code editor with a Python project. The Explorer sidebar on the left shows the project structure, including source files and test files. The main editor displays the `main.py` file with the following code:

```
9 print(new_customer)
10 print(hotel)
11
12 print("Reserve room")
13 reservation = hotel.reserve_room(new_customer)
14 print(reservation)
15 print(hotel)
16
17 print("Cancel reservation")
18 hotel.cancel_reservation(reservation)
19
20 print(hotel)
21
```

The bottom panel shows the TERMINAL output of the command `coverage report -m`, which displays a table of coverage statistics for all files in the project:

Name	Stats	Miss	Cover	Missing
src/_init_.py	0	0	100%	
src/customer.py	14	0	100%	
src/hotel.py	35	0	100%	
src/reservation.py	12	0	100%	
test/_init_.py	0	0	100%	
test/unit/_init_.py	0	0	100%	
test/unit/test_customer.py	22	0	100%	
test/unit/test_hotel.py	43	0	100%	
test/unit/test_reservation.py	25	0	100%	
TOTAL	151	0	100%	

The status bar at the bottom right indicates the current file is `main.py` at line 21, column 1, with 4 spaces, UTF-8 encoding, LF line endings, and Python 3.11.3 64-bit.