Pruebas de software y aseguramiento de la calidad

Profesor Titular:

Dr. Gerardo Padilla Zárate

Actividad

Actividad 6.2 | Ejercicio de Programación 3 y pruebas de unidad

Alumna

Andrea Carolina Treviño Garza A01034993

18 de Febrero del 2024

Actividad 6.2 – Ejercicio de programación 3 y pruebas de unidad

Después de crear el archivo “hotel_management.py” con las clases Hotel, Reservation & Customer con los métodos solicitados. Se procedió a analizar errores con Flake8 y Pylint.

Análisis de Errores usando Flake 8

En un inicio resultó así:

```
C:\Users\at60371>flake8 hotel_management.py
hotel_management.py:13:1: E302 expected 2 blank lines, found 1
hotel_management.py:33:80: E501 line too long (84 > 79 characters)
hotel_management.py:48:80: E501 line too long (84 > 79 characters)
hotel_management.py:71:80: E501 line too long (90 > 79 characters)
hotel_management.py:98:80: E501 line too long (86 > 79 characters)
hotel_management.py:103:80: E501 line too long (100 > 79 characters)
hotel_management.py:104:80: E501 line too long (92 > 79 characters)
hotel_management.py:118:80: E501 line too long (84 > 79 characters)
hotel_management.py:120:80: E501 line too long (85 > 79 characters)
hotel_management.py:123:80: E501 line too long (106 > 79 characters)
hotel_management.py:126:80: E501 line too long (90 > 79 characters)
hotel_management.py:139:80: E501 line too long (83 > 79 characters)
hotel_management.py:157:80: E501 line too long (86 > 79 characters)
hotel_management.py:167:80: E501 line too long (96 > 79 characters)
hotel_management.py:172:80: E501 line too long (96 > 79 characters)
hotel_management.py:175:80: E501 line too long (91 > 79 characters)
hotel_management.py:191:80: E501 line too long (91 > 79 characters)
hotel_management.py:207:28: W292 no newline at end of file
```

Se corrigieron errores:

- Agregar 2 espacios al inicio de una clase nueva.
- Al acortar extensión de las líneas de código marcadas.
- Al agregar una línea en blanco después de la última línea de código.

Al corregirlos así se vió:

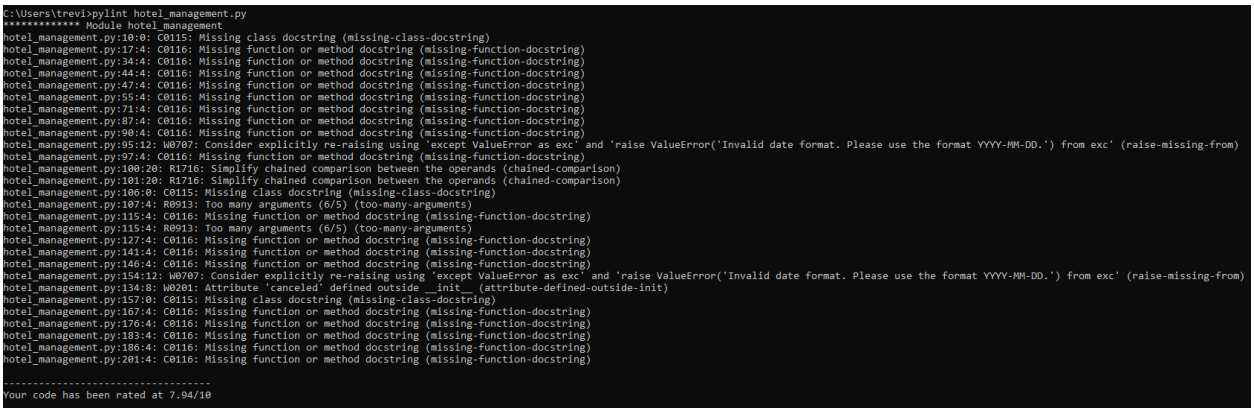
```
C:\Users\at60371>flake8 hotel_management.py

C:\Users\at60371>
```

Ya no desplegó ningún error en consola.

Análisis de Errores usando Pylint

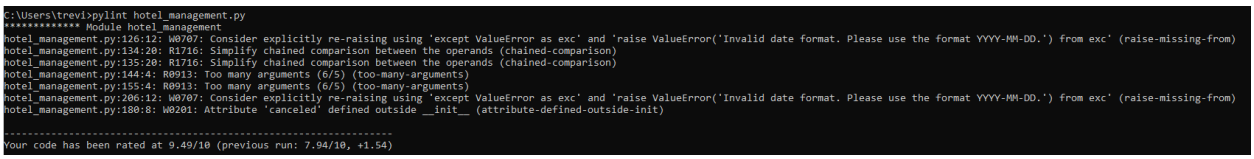
En un inicio se obtuvieron los siguientes errores y la siguiente calificación 7.94 (eso después de tener 0 errores en Flake8:



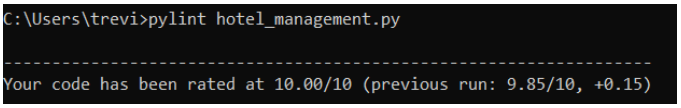
Se corrigieron errores:

- Se agregaron descripciones “docstrings” a todas las clases y sus respectivos métodos.
- Se agregó “from None” para aclarar excepción en error por fecha inválida ‘except ValueError as exc’.
- Se arreglaron comparaciones entre operandos en cadena, para que fuera más clara la comparación.
- Se redujo la cantidad de argumentos de las funciones que excedían cantidad.
- Se agregó atributo “.canceled” a función “__init__” de clase Reservation.
-

Después de arreglar varios de estos errores, la calificación de pylint fue de 9.49



Hasta finalmente llegar a 0 errores. Obteniendo calificación de 10.00



Casos de Prueba con “unittest” de Python:

Se crearon casos de prueba positivos en el archivo “test_hotel.py” para probar los todos los métodos de la clase “Hotel”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_hotel.py
test_cancel_reservation (test_hotel.TestHotelMethods.test_cancel_reservation) ... ok
test_cancel_room_reservation (test_hotel.TestHotelMethods.test_cancel_room_reservation) ... ok
test_create_reservation (test_hotel.TestHotelMethods.test_create_reservation) ... ok
test_display_information (test_hotel.TestHotelMethods.test_display_information) ... ok
test_modify_information (test_hotel.TestHotelMethods.test_modify_information) ... ok
test_reserve_room (test_hotel.TestHotelMethods.test_reserve_room) ... ok

----------------------------------------------------------------------
Ran 6 tests in 0.006s

OK
```

Además, se crearon casos de prueba negativos en el archivo “test_hotel_neg.py” para probar los varios tipos de error comunes en la clase “Hotel”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_hotel_neg.py
test_cancel_reservation_already_canceled (test_hotel_neg.TestHotelNegativeCases.test_cancel_reservation_already_canceled) ... ERROR
test_cancel_reservation_reservation_not_exists (test_hotel_neg.TestHotelNegativeCases.test_cancel_reservation_reservation_not_exists) ... ERROR
test_cancel_room_reservation_reservation_not_exists (test_hotel_neg.TestHotelNegativeCases.test_cancel_room_reservation_reservation_not_exists) ... ERROR
test_cancel_room_reservation_room_not_reserved (test_hotel_neg.TestHotelNegativeCases.test_cancel_room_reservation_room_not_reserved) ... ERROR
test_create_reservation_invalid_end_date (test_hotel_neg.TestHotelNegativeCases.test_create_reservation_invalid_end_date) ... ok
test_create_reservation_invalid_room_number (test_hotel_neg.TestHotelNegativeCases.test_create_reservation_invalid_room_number) ... ok
test_create_reservation_invalid_start_date (test_hotel_neg.TestHotelNegativeCases.test_create_reservation_invalid_start_date) ... ok
test_create_reservation_no_available_rooms (test_hotel_neg.TestHotelNegativeCases.test_create_reservation_no_available_rooms) ... ok
test_reserve_room_already_reserved (test_hotel_neg.TestHotelNegativeCases.test_reserve_room_already_reserved) ... ok
test_reserve_room_invalid_end_date (test_hotel_neg.TestHotelNegativeCases.test_reserve_room_invalid_end_date) ... ok
test_reserve_room_invalid_start_date (test_hotel_neg.TestHotelNegativeCases.test_reserve_room_invalid_start_date) ... ok
test_reserve_room_no_available_rooms (test_hotel_neg.TestHotelNegativeCases.test_reserve_room_no_available_rooms) ... ok
```

Aquí se ve el detalle de las 4 pruebas que fallaron:

```
======================================================================
ERROR: test_cancel_reservation_already_canceled (test_hotel_neg.TestHotelNegativeCases.test_cancel_reservation_already_canceled)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_hotel_neg.py", line 48, in test_cancel_reservation_already_canceled
    self.reservation.cancel_reservation()
  File "C:\Users\trevi\hotel_management.py", line 184, in cancel_reservation
    self.hotel.cancel_reservation(self)
  File "C:\Users\trevi\hotel_management.py", line 52, in cancel_reservation
    raise ValueError("Reservation is already canceled")
ValueError: Reservation is already canceled

======================================================================
ERROR: test_cancel_reservation_reservation_not_exists (test_hotel_neg.TestHotelNegativeCases.test_cancel_reservation_reservation_not_exists)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_hotel_neg.py", line 44, in test_cancel_reservation_reservation_not_exists
    self.assertFalse(self.hotel.cancel_reservation(reservation))
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\trevi\hotel_management.py", line 49, in cancel_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

======================================================================
ERROR: test_cancel_room_reservation_reservation_not_exists (test_hotel_neg.TestHotelNegativeCases.test_cancel_room_reservation_reservation_not_exists)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_hotel_neg.py", line 75, in test_cancel_room_reservation_reservation_not_exists
    self.assertFalse(self.hotel.cancel_room_reservation(reservation))
                     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\trevi\hotel_management.py", line 98, in cancel_room_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

======================================================================
ERROR: test_cancel_room_reservation_room_not_reserved (test_hotel_neg.TestHotelNegativeCases.test_cancel_room_reservation_room_not_reserved)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_hotel_neg.py", line 80, in test_cancel_room_reservation_room_not_reserved
    self.assertTrue(self.hotel.cancel_reservation(reservation))
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\trevi\hotel_management.py", line 49, in cancel_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

----------------------------------------------------------------------
Ran 12 tests in 0.024s
```

Pero despliegan correctamente el mensaje esperado para la prueba donde se despliega que la reservación se canceló o no existe. Lo que orientaría al usuario sobre como corregir el error.

Se crearon casos de prueba positivos en el archivo “test_reservation.py” para probar los todos los métodos de la clase “Reservation”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_reservation.py
test_cancel_reservation (test_reservation.TestReservationMethods.test_cancel_reservation) ... ERROR
test_create_reservation (test_reservation.TestReservationMethods.test_create_reservation) ... ok
test_is_canceled (test_reservation.TestReservationMethods.test_is_canceled) ... ERROR
```

Aquí se ve el detalle de las 2 pruebas que fallaron:

```
======================================================================
ERROR: test_cancel_reservation (test_reservation.TestReservationMethods.test_cancel_reservation)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_reservation.py", line 28, in test_cancel_reservation
    success, message = self.reservation.cancel_reservation()
                       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\trevi\hotel_management.py", line 184, in cancel_reservation
    self.hotel.cancel_reservation(self)
  File "C:\Users\trevi\hotel_management.py", line 49, in cancel_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

======================================================================
ERROR: test_is_canceled (test_reservation.TestReservationMethods.test_is_canceled)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_reservation.py", line 35, in test_is_canceled
    self.reservation.cancel_reservation()
  File "C:\Users\trevi\hotel_management.py", line 184, in cancel_reservation
    self.hotel.cancel_reservation(self)
  File "C:\Users\trevi\hotel_management.py", line 49, in cancel_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

----------------------------------------------------------------------
Ran 3 tests in 0.003s

FAILED (errors=2)
```

Pero despliegan correctamente que no existe la reservación cuando está cancelada.

Además, se crearon casos de prueba negativos en el archivo “test_reservation_neg.py” para probar los varios tipos de error comunes en la clase “Reservation”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_reservation_neg.py
test_cancel_reservation_already_canceled (test_reservation_neg.TestReservationNegativeCases.test_cancel_reservation_already_canceled) ... ERROR
test_create_reservation_invalid_end_date (test_reservation_neg.TestReservationNegativeCases.test_create_reservation_invalid_end_date) ... ok
test_create_reservation_invalid_start_date (test_reservation_neg.TestReservationNegativeCases.test_create_reservation_invalid_start_date) ... ok
```

Aquí se ve el detalle de la prueba que falló:

```
======================================================================
ERROR: test_cancel_reservation_already_canceled (test_reservation_neg.TestReservationNegativeCases.test_cancel_reservation_already_canceled)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\trevi\test_reservation_neg.py", line 34, in test_cancel_reservation_already_canceled
    self.reservation.cancel_reservation()
  File "C:\Users\trevi\hotel_management.py", line 184, in cancel_reservation
    self.hotel.cancel_reservation(self)
  File "C:\Users\trevi\hotel_management.py", line 49, in cancel_reservation
    raise ValueError("Reservation does not exist")
ValueError: Reservation does not exist

----------------------------------------------------------------------
Ran 3 tests in 0.008s

FAILED (errors=1)
```

Pero despliega nuevamente correctamente que no existe la reservación cuando está cancelada.

Finalmente se crearon casos de prueba positivos en el archivo “test_customer.py” para probar los todos los métodos de la clase “Customer”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_customer.py
test_create_customer (test_customer.TestCustomerMethods.test_create_customer) ... ok
test_delete_customer (test_customer.TestCustomerMethods.test_delete_customer) ... ok
test_display_information (test_customer.TestCustomerMethods.test_display_information) ... ok
test_modify_information (test_customer.TestCustomerMethods.test_modify_information) ... ok

----------------------------------------------------------------------
Ran 4 tests in 0.002s
```

Además, se crearon casos de prueba negativos en el archivo “test_customer_neg.py” para probar los varios tipos de error comunes en la clase “Customer”, aquí se observa el resultado:

```
C:\Users\trevi>python -m unittest -v test_customer_neg.py
test_create_customer_invalid_email (test_customer_neg.TestCustomerNegativeCases.test_create_customer_invalid_email) ... ok
test_create_customer_missing_information (test_customer_neg.TestCustomerNegativeCases.test_create_customer_missing_information) ... ok
test_delete_customer_not_exist (test_customer_neg.TestCustomerNegativeCases.test_delete_customer_not_exist) ... ok
test_modify_information_customer_not_exist (test_customer_neg.TestCustomerNegativeCases.test_modify_information_customer_not_exist) ... ok
test_modify_information_invalid_email (test_customer_neg.TestCustomerNegativeCases.test_modify_information_invalid_email) ... ok

----------------------------------------------------------------------
Ran 5 tests in 0.003s

OK
```

***Todos las pruebas negativas que salen como “ok” en las clases de “Hotel”, “Reservation” y “Customer” tienen mensajes de error definidos para manejar correctamente cada prueba.**

Bibliografía:

Flake8: Your Tool For Style Guide Enforcement — flake8 7.0.0 documentation. (s. f.). https://flake8.pycqa.org/en/latest/

Python Static Analysis Tools. (s. f.). Blog | Iamluminousmen. https://luminousmen.com/post/python-static-analysis-tools

unittest — Unit testing framework. (s. f.). Python Documentation. https://docs.python.org/3/library/unittest.html