



**Tecnológico
de Monterrey**

**Instituto Tecnológico de Estudios
Superiores de Monterrey**

**Maestría en Inteligencia Artificial Aplicada (MNA-
V)**

Pruebas de software y aseguramiento de la calidad

Nombre del trabajo:

6.2 Ejercicio de programación 3 y pruebas de unidad

Nombre del maestro:

Dr. Gerardo Padilla Zárate

Nombre del estudiante:

Kevin Brandon Cruz Mejia

Matricula:

A01794176

Fecha de entrega: 18 de febrero del 2024

Programa 1. Análisis de Errores de Pylint – PEP 8

Customer

```
(NLPSemana10) C:\Windows\system32>pylint "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\customer.py"
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

(NLPSemana10) C:\Windows\system32>
```

Hotel

```
(NLPSemana10) C:\Windows\system32>pylint "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\hotel.py"
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Reservation

```
(NLPSemana10) C:\Windows\system32>pylint "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\reservation.py"
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Room

```
(NLPSemana10) C:\Windows\system32>pylint "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\room.py"
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Todas las clases con 10 de puntuación

Programa 2. Análisis de Errores de Flake

Customer

```
(NLPSemana10) C:\Windows\system32>flake8 "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\customer.py"
(NLPSemana10) C:\Windows\system32>
```

Hotel

```
(NLPSemana10) C:\Windows\system32>flake8 "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\hotel.py"
(NLPSemana10) C:\Windows\system32>
```

Reservation

```
(NLPSemana10) C:\Windows\system32>flake8 "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\reservation.py"
(NLPSemana10) C:\Windows\system32>
```

Room

```
(NLPSemana10) C:\Windows\system32>flake8 "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\room.py"
(NLPSemana10) C:\Windows\system32>
```

Cero errores en las clases anteriores

Programa 1. Correcto Diseño de Casos de Prueba. Incluir casos negativos.

Casos positivos

Hotel

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python Test/test_hotel.py
.....
-----
Ran 6 tests in 0.005s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>
```

Room

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python Test/test_room.py
....
-----
Ran 4 tests in 0.000s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>
```

Reservation

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python Test/test_reservation.py
...
-----
Ran 3 tests in 0.002s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>
```

Customer

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python Test/test_customer.py
.....
-----
Ran 5 tests in 0.003s

OK
```

Casos negativos

Room

Primer caso

```
Room(room_number="hola", room_type="single", price=400.00, is_available=True)
```

Al cambiar el numero por hola al inicializar el cuarto con un numero que no es, el programa atrapa la excepción y te muestra cual es el error, por ello falla en las cuatro pruebas porque se inicializa mal desde el principio

```
Traceback (most recent call last):
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Test\test_room.py", line 13, in setUp
    self.room = Room(room_number="hola", room_type='single', price=400.00, is_available=True)
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\room.py", line 21, in __init__
    raise ValueError("El numero debe ser un entero")
ValueError: El numero debe ser un entero

-----
Ran 4 tests in 0.001s

FAILED (errors=4)
```

Segundo Caso

```
Room(room_number=101, room_type=111, price=400.00, is_available=True)
```

Al cambiar el roomtype por numeros al inicializar el cuarto con un tipo que no es, el programa atrapa la excepción y te muestra cual es el error, por ello falla en las cuatro pruebas porque se inicializa mal desde el principio

```
Traceback (most recent call last):
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Test\test_room.py", line 13, in setUp
    self.room = Room(room_number=101, room_type=111, price=400.00, is_available=True)
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\room.py", line 18, in __init__
    raise ValueError("El room type debe tener letras")
ValueError: El room type debe tener letras

-----
Ran 4 tests in 0.001s

FAILED (errors=4)
```

Tercer Caso

```
Room(room_number=101, room_type='single', price="Hola", is_available=True)
```

Al cambiar el precio por hola al inicializar el cuarto con un precio que no es, el programa atrapa la excepción y te muestra cual es el error, por ello falla en las cuatro pruebas porque se inicializa mal desde el principio

```
Traceback (most recent call last):
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Test\test_room.py", line 13, in setUp
    self.room = Room(room_number=101, room_type='single', price="Hola", is_available=True)
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\room.py", line 15, in __init__
    raise ValueError("El precio debe ser un entero")
ValueError: El precio debe ser un entero

-----
Ran 4 tests in 0.001s

FAILED (errors=4)
```

Customer

Cuarto Caso

```
Customer(customer_id=1, name=111, email='kevincruz@example.com')
```

Al cambiar el nombre por letras al inicializar el customer con un nombre que no es, el programa atrapa la excepción y te muestra cual es el error, por ello falla en las cinco pruebas porque se inicializa mal desde el principio

```
Traceback (most recent call last):
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Test\test_customer.py", line 16, in setUp
    self.customer = Customer(customer_id=1, name=111, email='kevincruz@example.com')
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\customer.py", line 13, in __init__
    raise ValueError("El nombre debe tener letras")
ValueError: El nombre debe tener letras

-----
Ran 5 tests in 0.001s

FAILED (errors=5)
```

Hotel

Quinto Caso

```
name = 1111
location = "Mexico City"
hotel = Hotel.create_hotel(name, location)
```

Al cambiar el nombre por números al inicializar el hotel con un tipo que no es, el programa atrapa la excepción y te muestra cual es el error, por ello falla en una prueba porque se encontró con error en el método de crear el hotel

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python Test/test_hotel.py
.E....
=====
ERROR: test_create_hotel (__main__.TestHotel)
Testing for creating a hotel correctly initializes its attributes.
=====
Traceback (most recent call last):
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Test\test_hotel.py", line 23, in test_create_hotel
    hotel = Hotel.create_hotel(name, location)
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\hotel.py", line 47, in create_hotel
    hotel = Hotel(name, location)
  File "C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3\Scripts\hotel.py", line 20, in __init__
    raise ValueError("El nombre debe contener letras")
ValueError: El nombre debe contener letras
=====
Ran 6 tests in 0.005s

FAILED (errors=1)
```

Este criterio depende de una competencia de aprendizaje Programa 1. Cobertura de líneas por clase

Clase Customer 97%

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python -m coverage run -m unittest Test/test_customer.py
....
Ran 5 tests in 0.004s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>coverage report -m
Name                               Stmts  Miss  Cover   Missing
-----
Scripts\customer.py                 37      1    97%      13
Test\test_customer.py              53     10    81%    66-78
-----
TOTAL                               90     11    88%
```

Clase Hotel 85%

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python -m coverage run -m unittest Test/test_hotel.py
.....
Ran 6 tests in 0.006s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>coverage report -m
Name                               Stmts  Miss  Cover   Missing
-----
Scripts\hotel.py                    59      9    85%    36-40, 86, 102-104
Scripts\reservation.py              26      5    81%    33-34, 39-41
Scripts\room.py                     23      5    78%    15, 18, 21, 35, 40
Test\test_hotel.py                  61     10    84%    82-94
-----
TOTAL                               169     29    83%
```

Clase Reservation 88%

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python -m coverage run -m unittest Test/test_reservation.py
...
Ran 3 tests in 0.003s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>coverage report -m
Name                               Stmts  Miss  Cover   Missing
-----
Scripts\reservation.py              26      3    88%    39-41
Test\test_reservation.py           34     10    71%    47-59
-----
TOTAL                               60     13    78%
```

Clase Room 88%

```
(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>python -m coverage run -m unittest Test/test_room.py
....
-----
Ran 4 tests in 0.000s

OK

(NLPSemana10) C:\Users\Professional\NoteBook\ITESM\Calidad de software\Ejercicio de programacion 3>coverage report -m
Name                               Stmts  Miss  Cover   Missing
-----
Scripts\room.py                     17      2    88%    34, 50
Test\test_room.py                   35     10    71%    41-53
-----
TOTAL                               52     12    77%
```

Los coverage anteriores solo son de las clases principales, no tome en cuenta los scripts que se hicieron de prueba test_customer, test_room, test_reservation, test_hotel, debido a que en la rúbrica menciona líneas por clase pero no las de tipo test tampoco en las instrucciones, así que asumi que solo en las clases de interés son las que se evalúan con las características PEP8, además de que todas las adecuaciones están en la clase.py como lo es PEP8, Fleak and Pylint por lo que no tome en cuenta en test_class.py para que tuviera estas características y mejorar el coverage para los scripts tipo test.