

**UNIVERSIDAD NACIONAL AGRARIA DE LA SELVA**  
**FACULTAD DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**



**INVESTIGACIÓN CIENTÍFICA**

**Docente** : **Brian Pando**

**Curso** : **Pruebas de Software**

**Integrante** : **Maria Mia Tauma Flores**

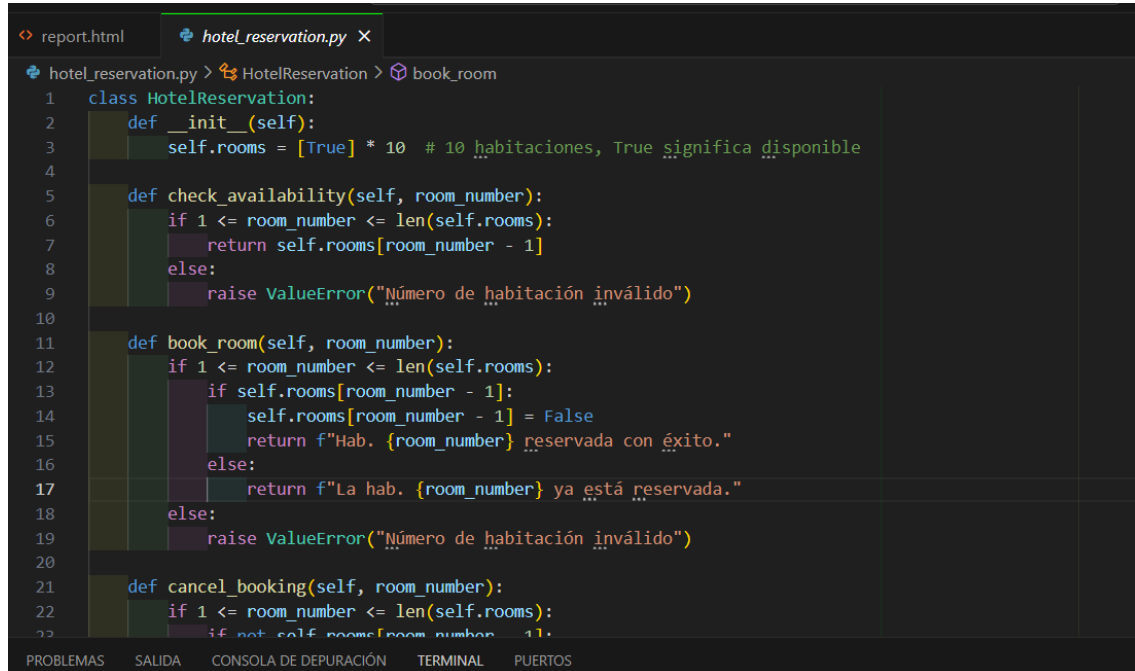
**Tingo Maria**

**2024**

Se dio petición del Docente que realice un Sistema de Reserva de un Hotel

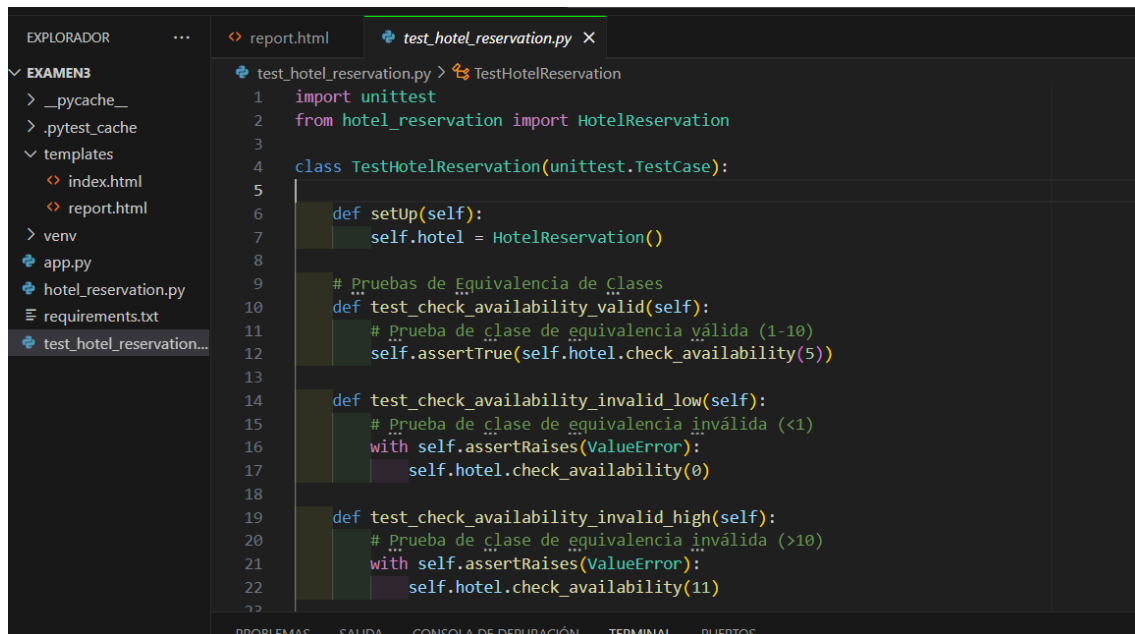
Realizando un Código Básico, de manera que en este se pudiera mostrar las 4 técnicas de caja negra que se fue realizando en clases.

Cree mi archivo “hotel\_reservation.py”



```
report.html hotel_reservation.py X
hotel_reservation.py > HotelReservation > book_room
1 class HotelReservation:
2     def __init__(self):
3         self.rooms = [True] * 10 # 10 habitaciones, True significa disponible
4
5     def check_availability(self, room_number):
6         if 1 <= room_number <= len(self.rooms):
7             return self.rooms[room_number - 1]
8         else:
9             raise ValueError("Número de habitación inválido")
10
11    def book_room(self, room_number):
12        if 1 <= room_number <= len(self.rooms):
13            if self.rooms[room_number - 1]:
14                self.rooms[room_number - 1] = False
15                return f"Hab. {room_number} reservada con éxito."
16            else:
17                return f"La hab. {room_number} ya está reservada."
18        else:
19            raise ValueError("Número de habitación inválido")
20
21    def cancel_booking(self, room_number):
22        if 1 <= room_number <= len(self.rooms):
23            if not self.rooms[room_number - 1]:
24                return f"Hab. {room_number} cancelada con éxito."
25            else:
26                return f"La hab. {room_number} no está reservada."
27        else:
28            raise ValueError("Número de habitación inválido")
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
```

Cree mi archivo “test\_hotel\_reservation.py”



```
EXPLORADOR ... report.html test_hotel_reservation.py X
test_hotel_reservation.py > TestHotelReservation
1 import unittest
2 from hotel_reservation import HotelReservation
3
4 class TestHotelReservation(unittest.TestCase):
5
6     def setUp(self):
7         self.hotel = HotelReservation()
8
9     # Pruebas de Equivalencia de Clases
10    def test_check_availability_valid(self):
11        # Prueba de clase de equivalencia válida (1-10)
12        self.assertTrue(self.hotel.check_availability(5))
13
14    def test_check_availability_invalid_low(self):
15        # Prueba de clase de equivalencia inválida (<1)
16        with self.assertRaises(ValueError):
17            self.hotel.check_availability(0)
18
19    def test_check_availability_invalid_high(self):
20        # Prueba de clase de equivalencia inválida (>10)
21        with self.assertRaises(ValueError):
22            self.hotel.check_availability(11)
23
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
```

Después de ello, creo mi reporte con el siguiente comando “pytest --html=templates/report.html --self-contained-html”

Este reporte facilita la comunicación con tu equipo o profesor, ya que puedes compartir el archivo HTML para mostrar lo que has hecho de manera clara y profesional. Finalmente, en un

entorno profesional, el informe es útil para procesos de integración continua, ayudando a asegurar que el código cumple con los requisitos antes de ser publicado o desplegado.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  JavaSE-21

* Debugger is active!
* Debugger PIN: 167-631-125

(venv) C:\Users\USER\OneDrive - Universidad Nacional Agraria de la Selva\1PRUEBAS DE SOFTWARE\examen3>pytest --html=templates/report.html --self-contained-html
===== test session starts =====
platform win32 -- Python 3.12.3, pytest-8.2.2, pluggy-1.5.0
rootdir: C:\Users\USER\OneDrive - Universidad Nacional Agraria de la Selva\1PRUEBAS DE SOFTWARE\examen3
plugins: html-4.1.1, metadata-3.1.1
collected 9 items

test_hotel_reservation.py ..... [100%]

- Generated html report: file:///C:/Users/USER/OneDrive%20-%20Universidad%20Nacional%20Agraria%20de%20la%20Selva/1PRUEBAS%20DE%20SOFTWARE/examen3/templates/report.html -
===== 9 passed in 0.02s =====

(venv) C:\Users\USER\OneDrive - Universidad Nacional Agraria de la Selva\1PRUEBAS DE SOFTWARE\examen3>
```

```
EXPLORADOR  ...  report.html X  test_hotel_reservation.py

EXAMEN3  [F]  [C]  [U]  [E]
> __pycache__
> .pytest_cache
✓ templates
  < report.html
  < report.html
> venv
  app.py
  hotel_reservation.py
  requirements.txt
  test_hotel_reservation...

templates > < report.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8"/>
5          <title id="head-title">report.html</title>
6          <style type="text/css">body {
7              font-family: Helvetica, Arial, sans-serif;
8              font-size: 12px;
9              /* do not increase min-width as some may use split screens */
10             min-width: 800px;
11             color: #999;
12         }
13
14         h1 {
15             font-size: 24px;
16             color: black;
17         }
18
19         h2 {
20             font-size: 16px;
21             color: black;
22         }
23
```

Despues del reporte hice otro archivo llamado “app.py” de manera que pueda ver las reservas de las habitaciones

```
1 from flask import Flask, render_template, request, redirect, url_for
2
3 app = Flask(__name__)
4
5 # Estructura de datos para almacenar las habitaciones
6 habitaciones = {
7     '101': {'nombre': '', 'reservada': False},
8     '102': {'nombre': '', 'reservada': False},
9     '103': {'nombre': '', 'reservada': False},
10    # Agrega más habitaciones según sea necesario
11 }
12
13 @app.route('/')
14 def index():
15     return render_template('index.html')
16
17 @app.route('/reserve', methods=['POST'])
18 def reserve():
19     nombre = request.form['name']
20     habitacion = request.form['room']
21     if habitacion in habitaciones and not habitaciones[habitacion]['reservada']:
22         habitaciones[habitacion]['nombre'] = nombre
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

(venv) C:\Users\USER\OneDrive - Universidad Nacional Agraria de la Selva\1PRUEBAS DE SOFTWARE\examen3>python app.py

Luego de ello entramos a nuestro localhost y podemos ingresar ciertos datos

Reserva de Hotel




127.0.0.1:5000

## Formulario de Reserva

Nombre:

Número de Habitación:

[Ver estado de las habitaciones](#)



Estado de las Habitaciones

×

+

←

↺

i

127.0.0.1:5000/habitaciones

# Estado de las Habitaciones

| Habitación | Nombre    | Estado       |
|------------|-----------|--------------|
| 101        | Maria Mia | Reservada    |
| 102        |           | No Reservada |
| 103        |           | No Reservada |