

# Actividad 6.2. Ejercicio de Programación

## 3

Juan Felipe Toro Salgado  
a01794247

### Descripción

Este ejercicio tiene como objetivo la implementación de un sistema de reservas de hoteles en Python siguiendo los requisitos especificados. El programa debe manejar hoteles, clientes y reservas, asegurando la persistencia de datos en archivos y la correcta validación de la información ingresada.

### Requisitos

#### Req 1: Implementación de Clases

El sistema debe implementar las siguientes clases en Python:

1. Hotel
2. Reservation
3. Customers

#### Req 2: Métodos Requeridos

El sistema debe incluir los siguientes métodos para manejar la persistencia de datos en archivos:

##### 1. Hoteles:

- Crear Hotel
- Eliminar Hotel
- Mostrar información del Hotel
- Modificar información del Hotel
- Reservar una habitación
- Cancelar una reserva

##### 2. Clientes:

- Crear Cliente
- Eliminar Cliente

- Mostrar información del Cliente
- Modificar información del Cliente

### **3. Reservas:**

- Crear una Reserva (Cliente, Hotel)
- Cancelar una Reserva

Los atributos de cada clase pueden ser definidos libremente siempre que permitan cumplir con los comportamientos requeridos.

## **Req 3: Implementación de Pruebas Unitarias**

Se deben implementar casos de prueba unitarios para cada método de las clases usando el módulo `unittest` de Python.

## **Req 4: Cobertura de Código**

El conjunto de pruebas unitarias debe lograr al menos un 85% de cobertura de líneas de código.

## **Req 5: Manejo de Errores**

El programa debe incluir mecanismos para manejar datos inválidos en los archivos de almacenamiento. Los errores deben mostrarse en la consola sin interrumpir la ejecución del programa.

## **Req 6: Cumplimiento con PEP8**

El código fuente debe cumplir con las normas de estilo PEP8.

## **Req 7: Validación con Herramientas de Calidad**

El código no debe mostrar advertencias al ser analizado con `flake8` y `pylint`.

## **Características Clave**

El programa debe incluir:

- Estructuras de control adecuadas.
- Entrada y salida por consola.
- Cálculos matemáticos cuando sean necesarios.
- Manejo de archivos para la persistencia de datos.
- Manejo de errores robusto.

## Cómo Usar el Código

1. Clonar el repositorio o descargar los archivos.
2. Instalar las dependencias necesarias ejecutando:
3. `pip install -r requirements.txt`
4. Ejecutar el programa principal:
5. `python main.py`
6. Para ejecutar las pruebas unitarias y verificar la cobertura:

```
python -m unittest discover tests/
```

```
coverage run -m unittest discover tests/
```

7. `coverage report -m`

8. Verificar cumplimiento con PEP8 y herramientas de calidad:

```
flake8 . > flake8_results.txt
```

9. `pylint hotel_reservation_system/ > pylint_results.txt`

## Evidencia de Ejecución

Se debe grabar la ejecución del programa y proporcionar los archivos utilizados en la tarea. La evidencia se encuentra en:

- Los resultados de `flake8` y `pylint` en los archivos `flake8_results.txt` y `pylint_results.txt`.
- Los reportes de cobertura generados en `unit_test.txt`.
- Los reportes en formato HTML almacenados en carpeta `html cov`.