



Actividad 6.2

Alumna:

Guadalupe Esmeralda González Maldonado (Ao1795767)

Maestría en Inteligencia Artificial Aplicada

**Materia: Pruebas de software y aseguramiento de la
calidad**

Profesor Titular: Dr. Gerardo Padilla Zárate

Febrero 2025

Contenido

ANÁLISIS DE FLAKE8.....3

Errores detectados y correcciones realizadas3

Conclusión de Flake8:.....3

MÉTODOS IMPLEMENTADOS4

CONCLUSIÓN5

Análisis de Flake8

Para garantizar la calidad y cumplimiento de los estándares de código en Python, se ejecutó Flake8 como herramienta de análisis estático en el archivo `Reservation_System.py`. A lo largo del proceso, se identificaron y corrigieron los siguientes aspectos:

Errores detectados y correcciones realizadas

1. Errores de espaciado y líneas en blanco (E302, E303)
 - a. Se ajustaron los espacios entre funciones y clases para cumplir con las reglas de PEP8.
 - b. Se eliminaron líneas en blanco innecesarias para mejorar la organización del código.
2. Errores de formato en líneas finales (W292, W391)
 - a. Se aseguró que el archivo finalice con una nueva línea, según lo recomendado por PEP8.
 - b. Se eliminaron espacios en blanco adicionales al final de líneas y bloques de código.
3. Líneas demasiado largas (E501)
 - a. Se refactorizaron líneas que excedían el límite de 100 caracteres, dividiéndolas en varias líneas para mejorar la legibilidad.
4. Implementación de la verificación automática de Flake8
 - a. Se incluyó la ejecución automática de Flake8 dentro del sistema de reservaciones.
 - b. Se configuró la generación de reportes automáticos en archivos `rflake8_report_<timestamp>.txt` para cada ejecución.

Conclusión de Flake8:

Tras la aplicación de las correcciones, el código cumple con los estándares de PEP8, lo que mejora su claridad, mantenibilidad y legibilidad. La integración de Flake8 en el sistema asegura una verificación continua de calidad y ayuda a prevenir errores de formato en futuras modificaciones.

Métodos Implementados

1. Gestión de Hoteles

- **Crear Hotel:** Se implementó el método `create_hotel()` para agregar un nuevo hotel y almacenarlo en `hotels.json`.
- **Eliminar Hotel:** `delete_hotel()` permite eliminar un hotel por su ID y actualizar el archivo JSON.
- **Mostrar Información del Hotel:** `display_hotels()` recupera y muestra la lista de hoteles almacenados.
- **Modificar Información del Hotel:** `modify_hotel()` permite actualizar el nombre y la ubicación de un hotel.
- **Reservar una Habitación:** Este comportamiento está implícito en la creación de una reservación.
- **Cancelar una Reservación:** `cancel_reservation()` permite eliminar una reservación de la lista y actualizar `reservations.json`.

2. Gestión de Clientes

- **Crear Cliente:** `create_customer()` agrega un nuevo cliente y lo almacena en `customers.json`.
- **Eliminar Cliente:** `delete_customer()` permite eliminar un cliente por su ID y actualizar el archivo JSON.
- **Mostrar Información del Cliente:** `display_customers()` recupera y muestra la lista de clientes almacenados.
- **Modificar Información del Cliente:** `modify_customer()` permite actualizar el nombre y correo electrónico de un cliente.

3. Gestión de Reservaciones

- **Crear una Reservación:** `create_reservation()` permite asignar una reservación a un cliente en un hotel y almacena la información en `reservations.json`.
- **Cancelar una Reservación:** `cancel_reservation()` elimina una reservación específica y actualiza el archivo JSON.

Conclusión

Tras la aplicación de las correcciones, el código cumple con los estándares de PEP8, lo que mejora su claridad, mantenibilidad y legibilidad. La integración de Flake8 en el sistema asegura una verificación continua de calidad y ayuda a prevenir errores de formato en futuras modificaciones. Además, los métodos implementados cumplen con los requisitos especificados en Req 2, asegurando la persistencia y gestión de hoteles, clientes y reservaciones de manera eficiente.