

Análisis de eficiencia BigO, Theta y Omega.

- En el peor caso (Big-O), la eficiencia es $O(n)$, donde n es el número de habitaciones, ya que los métodos que buscan y reservan habitaciones recorren toda la lista.
- En promedio (Theta), la eficiencia es $\Theta(n)$ porque la mayoría de los métodos necesitan recorrer todas las habitaciones.
- En el mejor caso (Omega), la eficiencia es $\Omega(1)$ para las operaciones de inicialización y cuando la habitación que buscamos está al principio de la lista.

En el método `__init__` de `Habitacion`, la eficiencia es $O(1)$, ya que solo inicializa las variables de una instancia, realiza un conjunto fijo de operaciones sin importar cuántas habitaciones haya. En los métodos `reservar` y `cancelar_reserva` de `Habitacion`, la eficiencia también es $O(1)$ porque solo comprueban una condición y cambian valores de la habitación.

El método `generar_habitaciones` en `Hotel` crea 15 habitaciones (3 pisos con 5 habitaciones cada uno), y para cada habitación realiza operaciones de $O(1)$, por lo que el tiempo que toma no cambia con la cantidad de habitaciones.

El método `mostrar_disponibilidad` en `Hotel` revisa todas las habitaciones para ver cuáles están disponibles, por lo que tiene una eficiencia de $O(n)$, donde n es el número de habitaciones (15). Los métodos `realizar_reserva` y `cancelar_reserva` en `Hotel` también recorren todas las habitaciones hasta encontrar la que buscan, lo que da una eficiencia de $O(n)$ en el peor caso, ya que puede necesitar revisar cada habitación una vez. El método `mostrar_info` en `Hotel` recorre la lista completa de habitaciones, por lo que tiene una eficiencia de $O(n)$ en el peor de los casos.

En el método `generar_grafico_ocupacion` en `Hotel`, se utiliza una lista que cuenta cuántas habitaciones están disponibles y cuántas reservadas. Esta lista recorre todas las habitaciones, por lo que su eficiencia es $O(n)$. En el bucle principal `main`, la eficiencia dependerá de qué opciones elija el usuario. Sin embargo, cada acción en el bucle tiene como máximo una eficiencia de $O(n)$, excepto los métodos de inicialización que son $O(1)$.