

EJERCICIO 17 – ALQUILER DE PROPIEDADES

Se desea diseñar e implementar una plataforma para gestión de reservas de propiedades que llamaremos OOBnB. En la misma, los usuarios pueden gestionar sus inmuebles para su alquiler así como también realizar reservas sobre estos.

De los usuarios se conoce el nombre, la dirección y el DNI. Cada usuario posee propiedades que desea alquilar, de las cuales se guarda la dirección, un nombre descriptivo y el precio que se desea cobrar por noche. Además, los usuarios pueden realizar reservas sobre cualquiera de las propiedades disponibles.

Nos piden implementar la siguiente funcionalidad:

- **Consultar la disponibilidad de una propiedad en un período específico:** dada una propiedad, una fecha inicial y una fecha final, se debe determinar si la propiedad está disponible el período indicado.
- **Crear una reserva:** Un usuario puede realizar una reserva para un período de tiempo determinado. Si la propiedad está disponible, se crea la reserva y la propiedad pasa a estar ocupada durante ese período. Si no lo está, la reserva no será creada.
- **Calcular el precio de una reserva:** Dada una reserva, se debe poder calcular su precio. El mismo se obtiene multiplicando la cantidad de noches por el precio por noche.
- **Cancelar una reserva:** Se debe permitir cancelar una reserva. En este caso, la propiedad pasa a estar disponible durante el período de tiempo indicado en la reserva. Esta operación sólo es permitida si el período de la reserva no está en curso.
- **Calcular los ingresos de un propietario:** Se debe calcular la retribución a un propietario, la cual es el 75% de la suma de precio totales de las reservas incluidas en un período específico de tiempo.

Notas sobre el diseño e implementación:

Para el manejo de los períodos de reserva se sugiere añadir un nuevo método a la interfaz `DateLapse` definida en el ejercicio anterior (ejercicio de Intervalos de tiempo).

A modo de sugerencia, la especificación del mismo puede ser la siguiente:

```
/**
```

```
Retorna true si el período de tiempo del receptor se superpone con el recibido por parámetro
```

```
**/
```

```
public boolean overlaps (anotherDateLapse: DateLapse)
```

Tareas:

- 1. Realice el diagrama del modelo conceptual.**
- 2. Realice el diagrama de clases UML**
- 3. Implemente en Java de la funcionalidad requerida.**
- 4. Implemente en JUnit pruebas automatizadas en donde:**
 - a) Se muestre cómo verificar la disponibilidad de una fecha para una propiedad que no tenga reservas en ese período.**
 - b) Se muestre cómo verificar la disponibilidad de una fecha para una propiedad en la que haya reservas en ese período.**
 - c) Se muestre cómo se reserva una propiedad para un período en particular.**
 - d) Se muestre cómo cancelar una reserva creada.**