Sistema de alquileres temporales

Informe

Programación Orientada a Objetos II

Profesores: Diego Cano - Diego Torres - Matias Butti

Integrantes:

- Lucas Coronel e-mail:lucas.j.coronel@gmail.com
- Nicolás Ploza e-mail: nicolasploza@gmail.com
- Alexander Ferragut e-mail: alexanderferragut@gmail.com

Fecha: 12/12/2024

Detalles de implementacion y patrones de diseño utilizados

PUBLICACIÓN Y ALQUILER

El sistema conoce a muchos usuarios y a muchos inmuebles. Asumimos que los Usuarios pueden ser inquilinos o propietarios(no pueden ser ambos a la vez). Los usuarios conocen su nombre completo, email y teléfono. Un usuario tambien conoce sus reservas, mientras los inquilinos conocen sus reservas pendientes y aceptadas, los propietarios solo conocen sus reservas pendientes. Cuando un propietario acepta una reserva pendiente, la saca de su lista de pendientes. Los inmuebles conocen todas sus reservas, incluyendo las canceladas ya que se considera que en un futuro se podría requerir información acerca de las reservas canceladas.
Un propietario puede dar de alta inmuebles en el sistema, el sistema valida que sean el tipo de inmueble y servicios aceptados.
Para modelar el estado de un inmueble se utilizó el patrón Object Type para los atributos, Servicios, fotos, forma de pago.
Para ciudad, superficie, país, dirección, check-in, check-out y capacidad se utilizaron tipos primitivos.
Respecto al precio, el inmueble cuenta con un precio default por día , el cual utiliza para calcular el precio en un periodo dado.
En caso de las fechas que pertenecen a un periodo específico declarado por el propietario, inmueble conoce a una clase Período la cual tiene una fecha de inicio y de fin y un costo por día.

BUSQUEDAS DE INMUEBLES

Para implementar la búsqueda de inmuebles se utilizó el patrón de diseño composite, se cuenta con una clase llamada Search, que consta de 3 filtros obligatorios la cual la cual interactua con la interfaz FiltroOpcional que es implementada por las clases And y Or que serían dos composite, los cuales se construyen a partir de dos objetos tipo FiltroOpcional. En cuanto a los leaf se tratan de filtros simples que tambien implementan la interfaz FiltroOpcional, donde cada uno implementa el metodo “cumpleCondicion” que se usa para el filtrado de inmuebles.
El Search tiene la funcionalidad de settear un FiltroOpcional y en caso de que este sea nulo no lo toma en cuenta para filtrar.

RANKING DE INMUEBLES Y PROPETARIOS

Tanto la clase Usuario como Inmueble implementan la interfaz Rankeable con lo cual saben recibir calificaciones, calcular los promedios de los puntajes por categoría y promedio de puntajes en total. Antes de recibir una calificacion se valida si la categoria corresponde a la entidad que se va a calificar usando al sistema quien es el que administra las categorias por los calificables.
La clase calificación tiene como atributo la "Categoría", que es un Object Type, y el puntaje para dicha categoría. Las calificaciones evitan que se pueda calificar a con nota mayor a 5 o menor a 1.

VISUALIZACION Y RESERVA

En el caso de esta funcionalidad el sistema tiene toda la lógica para brindarla mediante el uso de interfaz de usuario, el inmueble tiene todos los getters y tiene acceso a sus propietarios. En cuanto a los datos de propietarios, se accede mediante los getters de la clase Propietario, incluyendo su antigüedad en el sitio, cantidad de alquileres totales, promedios de puntaje, etc.

CONCRECION DE UNA RESERVA

La Reserva tiene como atributo el inmueble, al inquilino, fecha de inicio y de fin ,y el método de pago.

Cuando se reserva un inmueble, indicando el inquilino, fecha inicio, fecha fin y medio de pago, se realiza una validación para ver si el medio de pago es aceptado por el inmueble y se verifica si la fecha ingresada para reservar esta disponible para dicho inmueble (en caso de que no este disponible ver apartado “reserva condicional”). Luego se instancia una reserva cuyo estado es "Pendiente", se registra la reserva en el sistema, se agrega a la lista de reservas pendientes de propietario, a las reservas del inmueble y a la lista de reservas del inquilino.

El propietario tiene la responsabilidad de aceptar dicha reserva, cuando este lo acepta el estado de la reserva pasa a "Aceptada".

Cuando se acepta la reserva se elimina de la lista de pendientes del propietario y se le envía un correo al email del Inquilino mediante la Reserva usando la clase GestionadorDeNotificaciones.

ADMINISTRACION DE RESERVAS PARA INQUILINOS

La clase Usuario conoce todas sus reservas. En cuanto a la clase Inquilino puede también acceder a sus reservas de una ciudad particular, reservas futuras, las ciudades en donde tiene reservas.
Cuando el inquilino desee cancelar la reserva entonces reserva dispara un evento hacia su GestionadorDeNotificaciones el cual le hace llegar los eventos correspondientes a EmailSender, el cual esta interesado en el evento de la cancelación de una reserva.

ADMINISTRACION DEL SITIO

El sistema tiene la funcionalidad de dar de alta lo siguiente:
-Dar de alta Categoria: En cuanto a la categoria se uso un map cuya clave son los calificables (inquilino, propietario e inmueble) y sus categorias para cada uno.
-Dar de alta Tipo de Inmueble y servicios: se agregan al sistema los tipos de inmuebles y servicios que pueden ser aceptados por el sistema.
-Listados de gestión: la clase sistema tiene los métodos para responder dichos listados.

POLITICAS DE CANCELACION

Para modelar las políticas de cancelación se optó por un strategy donde el rol del strategy lo tiene la interfaz PoliticaDeCancelacion donde las concreteStrategy son: Cancelacion, SinCancelacion, Intermedia. El cálculo del costo para cada política de cancelación lo realizan estas.
Por otro lado, el inmueble contiene una de estas estrategias concretas y el calculo del costoDeCancelacion(): consiste en una delegación a la estrategia que tenga el inmueble para calcularlo.

NOTIFICACIONES

Registramos en esta parte a dos clases que notifican diferentes eventos, lo administramos con una clase manager llamada GestionadorDeNotificaciones, la cual dispara los eventos:

- Cancelación de un inmueble
- Reserva de un inmueble
- Baja de precio de un inmueble

Las clases que usan al gestionador para disparar estos eventos son Reserva e Inmueble, y las clases interesadas en estos eventos son AppMobile y SitioWeb, ambas a su vez implementan una interfaz para realizar las acciones

correspondientes. En este apartado tenemos en cuenta que en un futuro tanto el SitioWeb como la AppMobile puede cambiar su interés en algún evento.

RESERVA CONDICIONAL

Cuando se le envía una solicitud de reserva al inmueble y este no se encuentre disponible para la fecha solicitada, guardará la reserva en su lista de reservas condicionales. Si se cancela una de sus reservas, el inmueble procesa su lista de reservas condicionales y si encuentra alguna que esté dentro de la fecha que se canceló la saca de su lista de condicionales, la agrega a su lista de reservas y la envía al propietario para que la acepte.

Asumimos que una propiedad está reservada cuando se envía una solicitud de reserva al inmueble y este puede estar o no aceptada. Esto lo hacemos para que al propietario no le lleguen dos reservas pendientes en la misma fecha.

PATRONES DE DISEÑO UTILIZADOS

ROLES PARA STRATEGY:
Context: Inmueble
Strategy: PoliticaDeCancelacion
ConcreteStrategyA: Cancelacion
ConcreteStrategyB: SinCancelacion
ConcreteStrategyC: Intermedia

ROLES PARA COMPOSITE:
Client: Search
Component: FiltroOpcional
Composite: And - Or
Leaf: FiltroCantidadHuespedes - FiltroPrecioMinimo - FiltroPrecioMaximo

ROLES PARA STATE:
Context: Reserva
State: EstadoReserva
ConcreteStateA: Pendiente
ConcreteStateB: Aceptada
ConcreteStateC: Cancelada

ROLES PARA SINGLETON:
Singleton: Pendiente - Aceptada - Cancelada

ROLES PARA OBSERVER:
ConcreteSubject: Reserva - Inmueble
Subject(ChangeManager): GestionadorDeNotificaciones
Observer: Interesado
ConcreteObserver: SitioWeb - AppMobile - EmailSender