TP Final Programación Orientada a Objetos

1° Cuatrimestre 2021

Integrantes:

- Coria Lautaro: lautaroCoria97@gmail.com
- Segovia Ariel: ariel-110595@hotmail.com

Publicacion y Alquiler

Quizás la temática más simple en un comienzo hasta el momento de idear una estrategia para los diferentes precios que pueden existir para un inmueble dependiendo de las fechas elegidas por el propietario de dicho inmueble. Se delego la responsabilidad a una clase llamada PeriodoPrecio para que se encargue de guardar una determinada cantidad de fechas con su respectivo precio, el inmueble en si cuenta con un atributo precioPorDefecto, el cual se retorna si el dia solicitado no existe entre los periodos creados por el propietario. Por otro, lado la publicacion de dicho un inmueble se encuentra bajo la responsabilidad del sitio en si, que determina si es apto para sumarse a la lista de inmuebles, teniendo en cuenta el tipo del inmueble y los servicios.

Busqueda

Para la búsqueda decidimos utilizar el patrón Composite. El comportamiento en común estará en la interfaz IBusqueda, que será implementada por todos sus buscadores. La clase BuscadorCompuesto será nuestra clase compuesta, la cual se encargará de hacer la recursión con la lista de buscadores que haya en ese momento. Al instanciar un BuscadorCompuesto, se le agregan 3 parámetros que serán la ciudad, la fecha de entrada y la fecha de salida. Esto sucede porque en la consigna aclara que para realizar una búsqueda, estos 3 parámetros serán obligatorios, por lo tanto, con estos parametros se agregaran nuevos objetos a la lista de buscadores. A esta clase se le podrán agregar nuevos buscadores, que serán opcionales.

Ranking de Inmuebles y Propietarios

Se desarrollo un sistema basado en perfiles donde los usuarios pueden puntuar a los inmuebles y otros usuarios propietarios, siendo los primeros inquilinos que ya hubiesen alquilado y a los inquilinos por parte de los propietarios si ya la habian alquilado a estos. Los inmuebles y usuarios implementan la interfaz puntuable por estadia para poder recibir esta puntuacion la cual es guardada en las categorias que manejan estos mismos perfiles en su clase abstracta.

Visualizacion y Reserva

Tambien llevada a cabo a base de los perfiles, si bien la clase abstracta de estos perfiles manejan la puntuacion obtenida, las clases concretas brindan informacion del Usuario/Inmueble mas alla del puntaje. Cabe destacar que en el caso del inmuebel, contiene un atributo con el perfil de su propietario para manipular los comentarios tanto del propietario como del inmueble.

Concrecion de una Reserva

Las reservas se diseñaron en base al patron state donde se delegan la mayoria de sus metodos a estos estados. Siendo la clase reserva el contexto, la clase Estado el state, y las clases PendienteDeConfirmacion, Confirmado, Cancelado. La transicion entre dichos estados son llevadas a cabo estos mismos dependiendo del metodo que sea ejecutado.

Administracion de Reservas para inquilinos

Es una clase simple que solo realiza funciones sobre las reservas a cargo del inquilino o cancela las reservas. En este ultimo caso solo ejecutaria el disparador para dar inicia a dicha cancelacion de reserva.

Administracion del Sitio

Se creó una clase llamada Administrador Del Sitio que servirá para saber todos los inmuebles libres, la tasa de inmuebles, para dar de alta las categorías para el sitio (Propietario Inquilino e Inmueble) que luego serán calificadas, los servicios que serán seleccionados para publicar un inmueble y los tipos de inmueble que se utilizan en el sistema. Para saber los 10 inquilinos que más alquilaron, se creó un método que invoca a otro método que ordena la lista de todos los usuarios que alquilaron. El orden de la lista se hizo creando otra clase (CompararUsuarios) que implementa la interfaz Comparator. En esta clase se comparan las veces que alquilaron los usuarios, y así, con el método collections.sort podemos ordenar de mayor a menor, los inquilinos que más veces alquilaron.

Politicas de Cancelacion

Para poder cancelar una reserva se diseño esta misma (aparte de con el patron state para los estados) con el patron Strategy. Donde el mensaje cancelarReserva delega la responsabilidad de la cancelacion al concreteStrategy que este seteado con la reserva. La clase reserva cumple con el rol de contexto, la clase PoliticaDeCancelacion el rol de Strategy y las clases CancelacionGratuita, SinCancelacion y CancelacionIntermedia el rol de ConcreteStrategy.

A su vez se utiliza el patron Template Method para la implementacion del metodo cancelar(Reserva, LocalDate), el metodo diferenciaDeDiasEsMayor es el concrete AbstractClass operation y el valorPara es el hookMethod, el cual va a ser sobrescrito por las subclases.

Por otro lado la clase CancelacionIntermedia dispone de instancias de las clases AbonoTotal, SinAbono y AbonoDel50PorCiento que heredan de la clase Abono los cuales reciben la responsabilidad darle valor al metodo valorPara.

Notificaciones

Para resolver las notificaciones de los eventos que suceden en los inmuebles, decidimos utilizar el patrón Observer. Por un lado tenemos el objeto que será observado, la cual será la clase Inmueble. Y por otro los objetos que observaran al inmueble, que serán las clases SitioWeb, AppUser El objeto observado será la clase inmueble, que alertará a sus observadores en 3 ocasiones; La cancelación de un inmueble, la reserva de un inmueble y la baja de precio. Cuando cualquiera de estas suceda notificará a sus observadores con el metodo notificar(String) pasandole como parametro el nombre del evento que sucedió (estos pueden ser “Baja de precio” “Cancelación”, “Reserva”). Cuando los observadores sean notificados, preguntaran si el nombre del evento es el que les corresponde. En caso de que lo sea, se alertará en pantalla con un mensaje para cada observador. El sitio web está interesado en la baja de precio por lo tanto imprimirá en pantalla un mensaje de que X inmueble bajo de precio a $x precio. La aplicación del usuario está interesada en la cancelación de una reserva de inmueble, por lo tanto imprimirá en pantalla un mensaje de que X inmueble está libre.

Reservas Condicionales

Las reservas cuentan con un atributo colaDeReservas el cual es una lista de reservas. Esta lista se incrementa cuando el inmueble encola una reserva a las reservas que imposiblitan esta que se pasa por parametro. Cuando se cancela una reserva se dispara un mensaje que provoca que el inquilino correspondiente a la primer reserva de la cola de la reserva cancelada, solicite dicha reserva, esto en el caso que no haya otra reserva confirmada que la imposibilite, de ser asi, simplemente se borra de la cola.